JAAS Bolsters Java Security

Explore the db4o Database Engine

JAVAPro™

A Fawcette Technical Publication

September 2001 Vol. 5 No. 9

www.javapro.com

Downsize Your Apps for J2ME

Dynamic Web Pages Made Easy with Apache Xang

Create a Top-Notch Tokenizer

Discover the Genius of Generics in Java

Free Source Code

Code Online
www.javapro.com

U.S. $5.95 (Canada $7.95)

09>

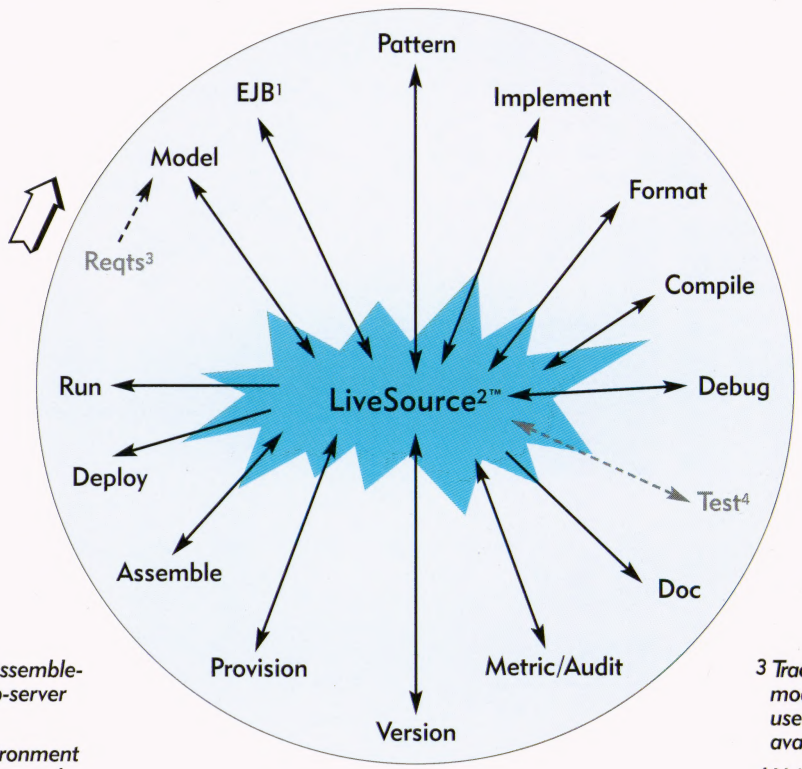
7 25274 91607 8

Experience Together ControlCenter 5 The #1 Model-Build-Deploy Platform

More than a modeling tool, more than an IDE, more than a deployer: Together ControlCenter provides integrated-on-a-single platform, deep integration from start to finish. Use it to model-pattern-edit-implement-compile-debug-version-doc-metric/audit-provision-assemble-deploy-run.

TOP NEW FEATURES IN VERSION 5

- Diagram-editor advances: new folded compartments, snap-to-grid, web-application diagrams, robustness diagrams
- Programming-editor advances: new deep-search usages, JSP and HTML syntax, javadoc code-completions, keyboard shortcuts for bookmarks
- JSP advances: new JSP/HTML editor; plus deployment and debugging with Tomcat
- Servlet advances: complete servlet modeling and generation
- EJB advances: new 2.0 support, EJB verification, works with over a dozen leading app-server products
- Audit and metrics: results to HTML; plus new C++ support
- XMI: full and complete coverage, in accordance with the Unisys standard.



1 EJB provision-assemble-deploy to 7 app-server products

2 The entire environment works with source code; always in-sync, always up-to-date.

3 Track requirements with model properties, or use the building block available for DOORS

4 Using the building block for JUnit.

Together ControlCenter 5

The #1 Model-Build-Deploy Platform

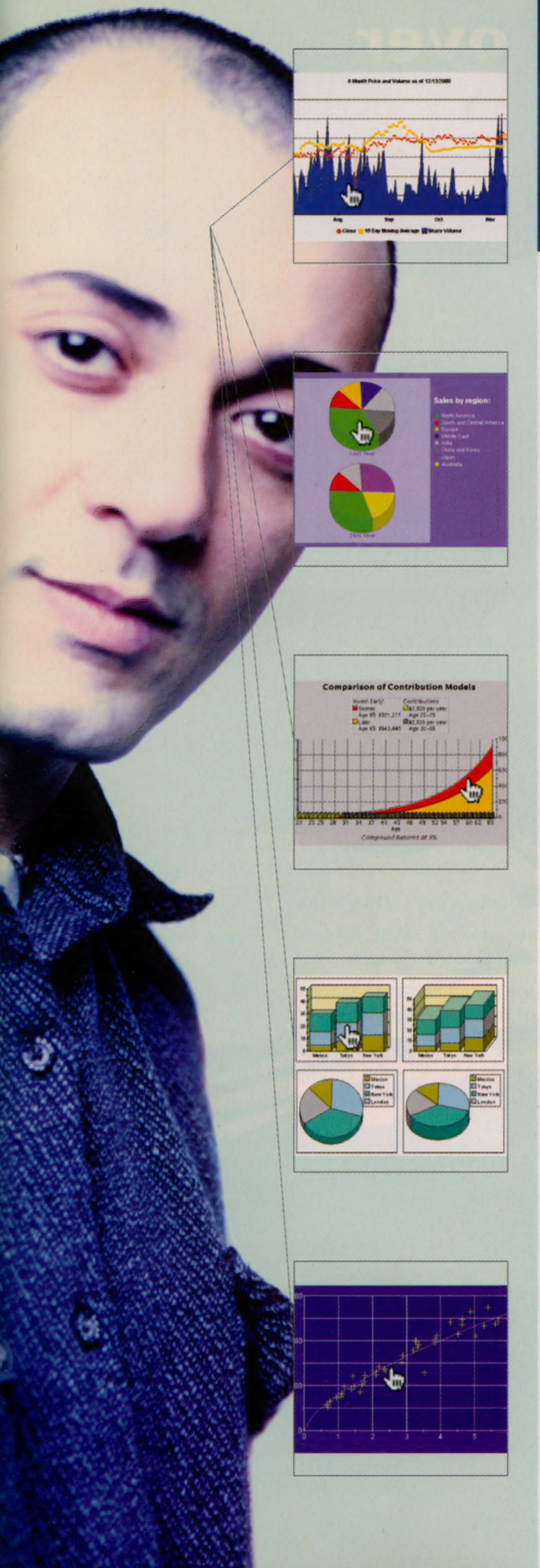
Tim Fielden at InfoWorld Test Center reports: "Business case: Using a single tool for app development fosters synergy between programmers and boosts their productivity. Coupling modeling capabilities with code development, TCC allows users to see the fruits of their efforts, both visually and textually. Technology case: TCC can deal with just about any technical challenge. Its adherence to the UML specification and its support for almost any development methodology make this a prime choice for shops that have yet to standardize with a software platform. Rating: Excellent"

Download Together ControlCenter 5 today and experience it for yourself. Use this special URL to extend your eval period 33% longer (20 days, rather than 15). www.togethersoft.com/jp

TogetherSoft Corporation. Improving the ways people work together.™ The adaptive business-process automation™ company. 1-919-833-5550 fax 1-919-833-5533 www.togethersoft.com. Offices in 10+ countries worldwide.

©2001 TogetherSoft Corporation. All rights reserved. Patents pending. Together is a registered trademark of TogetherSoft Corporation. TogetherSoft, Together ControlCenter, LiveSource, "Adaptive business-process automation," and "Improving the ways people work together" are trademarks of TogetherSoft Corporation.

TogetherSoft™



JClass ServerChart™

The **ultimate** charting component for **server-side** applications.

Introducing the first truly powerful solution for thin-client data visualization: JClass ServerChart. With a wide range of chart and graph types and built-in image map tag generation, JClass ServerChart gives end users of your browser-based apps access to business-critical data in an intuitive and highly interactive format.

- multiple business and scientific chart types
- integrates with all major application servers and IDEs
- works in headless environments when used with eTeks' PJA toolkit
- support for Windows 98/NT/2000/ME; Solaris 2.8, Redhat Linux 6.2, AIX 4.3.3
- support for JDK 1.1.8, 1.2.x, 1.3.x

sitraka software

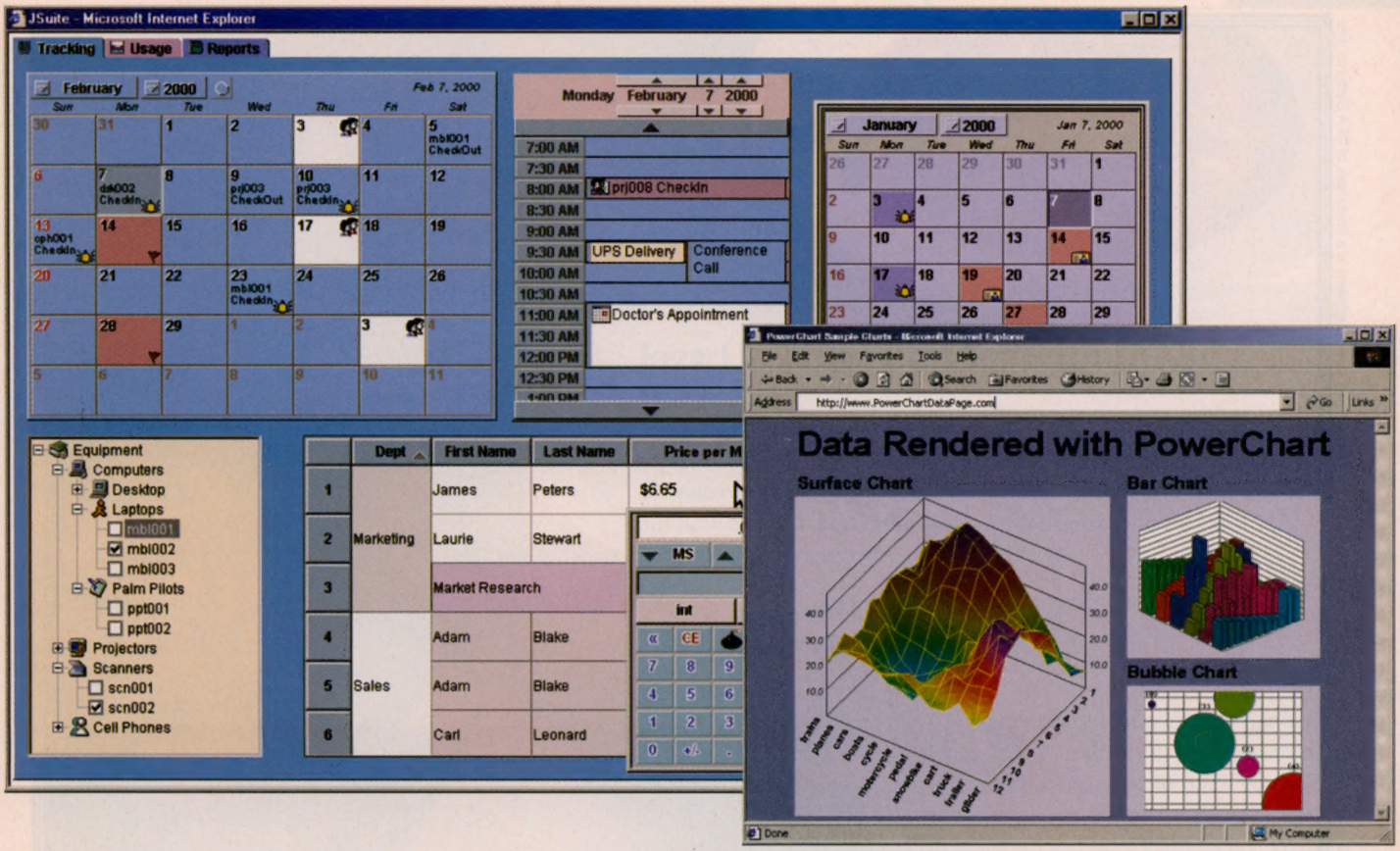
the Java™ advantage

Download a free eval now

www.sitraka.com/serverchart1/jp

1 888-361-3264 | Europe +31(0)20 510 67 00

Your SEARCH is over
the *right* Java toolset
the *right* support
Java • ActiveX • .NET



the *right* solution for you.

JFCSuite™ (Swing) and **JSuite™** (AWT/JavaBeans) provides the most comprehensive set of Java-based components available today. Deliver applications with the rich look and feel your users expect.

- Complete toolset includes grid, charting, explorer, tree, calendaring, data-input controls, calculator, color picker, and more...
- Light-weight for efficiency, heavy-weight in features
- Data Models™ for XML, JDBC, sockets, and URLs, including our exclusive AutoSynch™ feature, providing automatic data synchronization between our chart and grid
- Intuitive object model provides flexibility and control over an unprecedented selection of properties - adding professional polish to your application interface
- Server-side charting quickly delivers stunning data visualization. Our exclusive ActiveImage™ technology integrates with JSP and HTML pages to deliver interactive thin client charts using standard image map and JPEG files (requires J2EE on the server)

Enterprise Editions include the product, annual subscription service, GUARANTEED priority response for support, and many other special advantages

Corporate programs for large development teams

Backed by Infragistics' renowned quality, support and service

Infragistics - formed by the merger of Sheridan and ProtoView

For detailed information, visit
www.infragistics.com
Download Free Trial Version!
Order Online! or call 800-231-8588

Infragistics™
Component powered infrastructure

Copyright 2001 Infragistics, Inc. All rights reserved. Infragistics, the Infragistics logo, JFCSuite, JSuite,Data Models, ActiveImage & AutoSynch & PowerChart are trademarks of Infragistics, Inc.
All other trademarks or registered trademarks are the respective property of their owners.

©2001 BEA Systems, Inc. BEA and WebLogic are registered trademarks and How business becomes e-business is a trademark of BEA Systems, Inc.

With BEA WebLogic® you'll develop applications faster. So you can really get down to business.

The world's #1 e-business platform just got a lot more powerful. BEA WebLogic now has personalization and portals, application integration, customer management, e-commerce, B2B integration and business process management.

So now you can develop and integrate your applications in weeks instead of months.

Which means you might find that you'll have some extra time on your hands. How you spend it is, well, up to you.

Download your free trial copy of BEA WebLogic today at www.bea.com/download. And be sure to check out all of our free developer resources at **developer.bea.com.**

bea™

How business becomes e-business.™

JAVAPro™

COVER STORY

The Great Migration

by Daniel F. Savarese

Migrating code to the new (smaller) world created by J2ME is a challenge, but a necessary one: Consumers want to use the latest versions of their favorite applications on PDAs and mobile phones. Find out how to make the transition a smooth one.

GO ONLINE
www.javapro.com

- Download code for this issue and past issues
- Talk to the editors
- Join the Premier Club

SERVER

23 Jazz Up Java Security with JAAS

by Tarak Modi

A powerful combination of class loaders, security managers, and access controllers helps developers create secure applications, but Java is weaker in the areas of user authentication and authorization. Find out how the Java Authentication and Authorization Service (JAAS) enhances the Java security model.

39 COLUMN: JAVATECTURE
A Walk in the Clouds

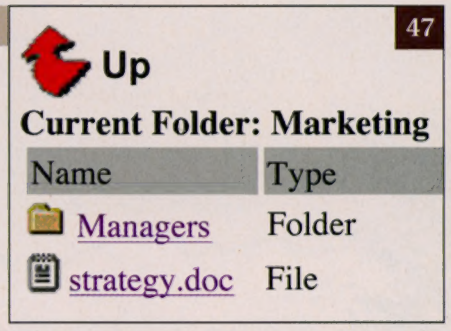
by James W. Cooper

A persistent problem with Java has been the lack of portability of the underlying system tools you need to build complete applications on either the client or the server. Find out how Cloudscape, an all-Java database system, can help you write a truly cross-platform user application.

WEB

47 Manage Documents on the Internet

by Budi Kurniawan



Wouldn't it be great if there were an Internet document-management system that had the same look-and-feel of your non-Web file system? Now there is. Check out Docman, an experimental project that uses JSP technology and is like a Windows Explorer for the Web.

65 COLUMN: WEBLICATION
The Zen of Xang

by Peter Varhol

Writing dynamically created Web pages based on user input is a necessity, but you need an approach that places as few dependencies as possible on the system architecture. Enter the Apache Xang interface, which just might be exactly what you're looking for.

Contents Continued

BPA INTERNATIONAL
PRINTED IN USA

Java Pro is published monthly—plus a special issue in September—for $34.97 per year, by Fawcette Technical Publications Inc., 209 Hamilton Avenue, Palo Alto, CA, USA, 94301-2500. Tel. 650-833-7100; Fax. 650-321-3818. **Postmaster:** Send address changes to *Java Pro*, P.O. Box 54584, Boulder, CO 80322-4584. Application to Mail at Periodicals Postage Rates is Pending at Palo Alto, CA and at additional mailing offices. **© 2001 Fawcette Technical Publications Inc.,** all rights reserved. *Java Pro* is an independent publication not affiliated with Sun Microsystems. Sun Microsystems is not responsible in any way for the editorial policy or other contents of the publication. All contents of *Java Pro* are copyright © 2001 by Fawcette Technical Publications Inc., unless otherwise noted. "VBITS" and "Interactive Developer" are trademarks of Fawcette Technical Publications Inc., a California Corporation, James E. Fawcette, President. Java is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication. **Canadian GST#: 86831 0764 ISSN: 1096-4495.**

The Network Is The Computer™ ©2001 Sun Microsystems, Inc. All rights reserved. Sun, Sun Microsystems, the Sun logo, Forte, iPlanet, all iPlanet based trademarks, NetBeans, Java, the Java coffee cup logo and The Network Is The Computer are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

build (n)

With Forte™ for Java™ tools, you can build Web services faster.

You just have to know when to stop.

Forte™ for Java™ Tools

You've heard it all your life: When you're building something—*anything*, for that matter—it's important to start with a solid foundation. If you're building Web services, the same rules apply. That's where Forte™ for Java™ tools come in. Forte tools are the all-powerful, easily extensible toolset for Sun™ Open Net Environment (Sun ONE), the software architecture behind the smartest services on the Web. They're tightly integrated into the iPlanet™ Application Server, so you can quickly test, fix and change your code *while your program is still running*. And because they're based on the NetBeans™ Tools Platform, you can use Forte tools to build your killer apps—with 100% pure Java code—on just about *any* platform around. IBM's tools? They only really work with IBM's platform, thereby limiting your innovation and, unfortunately, your options. And because Forte tools are open source, they're continually evolving with help from the million or so members of the developer community who've already downloaded a free copy. With that kind of upside, it's no surprise that Forte tools landed on the Crossroads A-List as the best open source Java IDE in the land. So if you're eager to speed time to market—and we know you are—Forte for Java tools will help you put up the smartest Web services on the block.

To download a free copy of Forte tools, go to www.sun.com/forte/ffj.

take it to the n[th]

Sun microsyste

ONLINE RESOURCES

Locator+ code: JP0109 Type this code in the Locator+ field at the top of the *Java Pro* home page. You can find all the code in the current issue of *Java Pro* at www.javapro.com. All the listings and associated files discussed in the articles are available free in one ZIP file. To access code in issues from 1998 and 1999, type "JP" followed by the month and year. For example, the Locator+ code for the October 1998 issue is JP1098. To access code from issues published after December 1999, type "JP" followed by the year (two digits) and the month (two digits). For example, the Locator+ code for the January 2001 issue is JP0101.

Premier Club Members: Members of the Premier Club of the Development Exchange (www.devx.com, $24.95 for six months, $44.95 for one year) can get each article's listings and associated files in separate ZIP files. For example, to download the individual code file associated with the October 1998 "Ask the Java Pro" column by Daniel F. Savarese, you would type in JP1098AP as the Locator+ code. For any articles published after December 1999, however, the position of the year (two digits) and the month (two digits) will be switched in the Locator+ code. For example, to download the individual code file associated with the January 2001 "Pro Shop" column, you would type in JP0101PS. Premier Club members can also access archives of all articles and related code ever published in *Java Pro*.

Java Zone: The Java Zone (www.java-zone.com) on the Development Exchange (www.devx.com) offers additional Java resources including product reviews, programming tips, weekly downloads, and discussion forums.

HOW TO REACH US

Editorial Offices: Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; Phone: 650-833-7100; Fax: 650-853-0230; Web: www.javapro.com; e-mail: java-pro@fawcette.com.

Article Submission: To submit articles for publication please contact Kay Keppler, Executive Editor, kkeppler@fawcette.com. Download the Author Guidelines at www.javapro.com/authguid.asp.

Product Reviews: To submit products for review please contact Kay Keppler, Executive Editor, kkeppler@fawcette.com.

Product Announcements: Please e-mail press releases to pressreleases@fawcette.com.

Reprints and Permissions: For all *Java Pro* editorial and advertising reprints contact PARS International Corp., 114 East 32nd Street, Suite 503, New York, NY 10016; Phone: 212-221-9595; Fax: 212-221-9195.

To Quote from an Article: Please contact Tami Stoeckle, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; Phone: 650-833-7118; E-mail: tstoeckle@fawcette.com. Specify the issue date and title of the article, the portion you would like to quote, and the purpose.

Photocopy Rights: Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Tami Stoeckle for more information.

Customer Service and Subscription Information: For subscription orders, inquiries, or address changes please contact Customer Service, *Java Pro*, P.O. Box 54584, Boulder, CO 80322-4584; Phone: 303-945-5282; Fax: 650-321-3818. Foreign and Canadian orders must be payable in U.S. dollars, plus postage. The surface rate to Canada and Mexico is $52.97 per year. For all other countries the air mail rate is $78.97 per year.

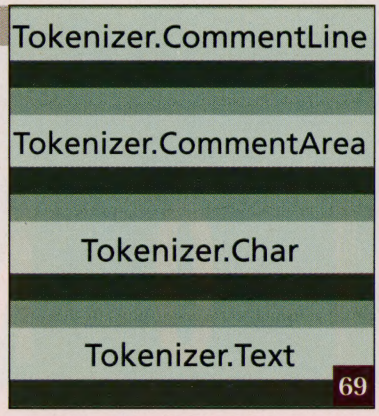
Back Issues: To order *Java Pro* back issues, call 650-833-7100 or 800-848-5523 and ask for Customer Service. Back issues cost $10. Additional postage will be charged to deliveries outside the USA.

CLIENT

69 A Tokenizer for Your Collection

by Claude Duguay

Numerous applications require parsing data but the tokenizer classes provided as part of the standard Java distribution fall short. FlexTokenizer is a powerful tokenizer that can be used to tokenize virtually any text source. See how it can work for you.



79 COLUMN: VISUAL COMPONENTS

Scratching the Surface

by Claude Duguay

Ready for a new look? The JSurface component manages panels with graphic borders and texture backgrounds. See how you can use it to redecorate your user interface with new visual effects.

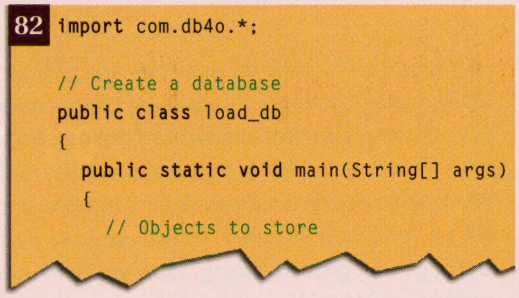
MOBILE

82 COLUMN: JAVA TO GO

OO Databases Get Small Minded

by Rick Grehan

The db4o database engine is remarkably small and is ideal for handheld applications. Discover the benefits and shortcomings of using such a simple database.



DEPARTMENTS

11 EDITOR'S NOTE
by Steve Gillmor

13 LETTERS

16 IN BRIEF

93 AD INDEX

96 PUBLIC STATIC
by Sean Gallagher

90 PRO SHOP
by Daniel F. Savarese

The generics extension to Java will be included in the JDK 1.5 release, but you can start programming with this essential language feature today. Learn how.

REVIEWS

14 INSTALLANYWHERE ENTERPRISE EDITION 4
A tool for creating cross-platform setup programs for both desktop and server applications

94 BOOK REVIEWS

IONA. JAVA.

FOUR LETTERS EACH.

COINCIDENCE?

OH, WE DON'T THINK SO.

IONA

iPortal Application Server™ 3.0

A J2EE conformant platform expressly designed to combine – out of the box – application serving and integration capabilities. With support for Web services built in, iPortal Application Server 3.0 is intended for organizations needing to connect middle-tier Java applications to enterprise applications running on other platforms inside the enterprise and across the Internet. And offering the fastest J2EE roundtrip engineering of any application server, along with a number of other very original developer productivity features, iPortal Application Server is custom-made by Extreme Programmers for Extreme Programmers.

Application servers are today's platform for new application development and deployment. They are also central to a number of dramatic information technology trends, the combined effects of which are driving the evolution of an entirely new generation of application server technology:

- The enterprise use of Java is exploding. The growing maturity of Sun's Java 2 Enterprise Edition (J2EE) specification and Enterprise JavaBean (EJB) component model makes it possible for IT organizations to build serious, large-scale Java applications. This in turn is placing increasing demands on application server vendors to provide dramatically higher scalability, reliability, and especially flexibility than was previously required.

- The anticipated business benefits of Internet-based process collaboration and business-to-business integration (B2Bi) are driving significant application integration investments in information-intensive organizations. These investments involve internal, application-to-application integration (A2Ai) projects, as well as external B2Bi projects. Organizations increasingly regard application servers as the appropriate platform on which to develop and host integration projects, which is leading these organizations to demand greater built-in application integration features in their application servers.

- A very significant new use of XML, Web services, promises to simplify – through standardization – application integration both within enterprises and across the Internet. At the same time, Web services offer the potential to extend the Internet's current two-tier architecture to a services-based, n-tier architecture. As the interest in Web services grows among IT organizations, so too has grown the demand for Web services-enabled application servers.

- All of these trends are acting to popularize the use of application servers. But as a broader population of developers attempts to adopt this powerful technology, ease of use becomes a critical factor. And in this era of enterprise applications running on an application server platform, ease of use means more than graphical development tools. Application servers must provide a convenience at deployment time that they have historically lacked.

- Popularization, and the encroachment of Microsoft with its .NET initiative, will also drive down the price of J2EE application servers, even as they offer unprecedented capabilities. This commodity pricing will have a severe impact on some vendors.

Specifications

- J2EE 1.2 Compliance
  - Servlet and JSP support
  - Full Entity Bean and Session Bean Support
  - JNDI Provision
  - Container and Bean managed persistence
  - 2-phase XA Transaction (JTA) support
  - Support for JMX based Management
  - JMS support
- Operating System Support
  - Windows NT/2000 on Intel
  - Solaris 8 on SPARC
  - HP/UX 11 on PA-RISC
  - AIX 4.3.3 on PowerPC
  - Tru64 5.1 on Alpha
  - Database Support
- JDBC 2.0 XA support for:
  - Oracle 7/8i
  - MS SQLServer 7 & 2000
  - UDB 5.2/6.1
  - Informix 9
  - Sybase 12

www.iona.com

Total Management

The IONA iPortal Application Server 3.0 includes iPortal Administration Console, using the latest JMX technology, provides a complete, comprehensive and extensible instrumentation and management solution for J2EE and existing legacy applications, all manageable from this single console, remotely over the web or through SNMP agents.

Enterprise Messaging

iPortal Application Server 3.0 ships with the award winning SonicMQ messaging product, offering an enterprise implementation of the JMS specification. Message Connection Pooling and Consumer classes deliver message-driven Bean and Servlet capabilities, improving J2EE application interoperability.

Application Interoperability

iPortal Application Server 3.0 itself interoperates with existing applications written in C++, Java, COM+, C#, vb.NET, COBOL and PL/1 running on Windows, Unix and Mainframe platforms. Allowing existing operational systems to be leveraged with J2EE applications in e-business systems protects existing investment.

Mainframe Integration

Unique to IONA is its ability to integrate existing OS/390 applications into object-based environments. iPortal Application Server extends this integration by allowing these mainframe applications to seamlessly interoperate with J2EE applications, while still preserving the transactional and security properties in which these applications operate.

Rapid Round-trip J2EE Development

By using a sophisticated source code structure, incremental build system and dynamic class loading technology, the iPortal Application Server 3.0 significantly improves application development time to market.

Wide Platform and Database Support

iPortal Application Server 3.0 offers the widest support of platform – including Windows NT, Windows 2000, HP-UX, Solaris, AIX and Tru64 – and database configurations – including Oracle, SQLserver, Cloudscape, Sybase, Informix, UDB and dB2—available.

The Core of the IONA e-Business Platform

iPortal Application Server 3.0 is at the core of IONA's e-business platform for Total Business Integration™, providing an end-to-end e-business integration solution combining A2Ai, B2Bi, XML, Web services, J2EE, CORBA and Enterprise Portal technologies.

To download your evaluation copy of iPortal Application Server 3.0, visit our download center at www.iona.com/downloads

IONA

Total Business Integration™

Corporate Headquarters
Dublin Office
The IONA Building
Shelbourne Road
Dublin 4 Ireland
t:+353 1 6625255
f:+353 1 6625244

US Headquarters
Waltham Office
200 West Street 4th Floor
Waltham,MA 02451
t:+1 781 9028000
f:+1 781 9028001

Santa Clara Development Center
2350 Mission College Blvd
Suite 650
Santa Clara,CA 95054
t:+1 408 3509500
f:+1 408 3509501

IONA

HIS IS IONA iPORTAL APPLICATION SERVER™ 3.0

FULL J2EE CERTIFICATION AND CONFORMITY

VIRTUALLY LIMITLESS FLEXIBILITY AND SCALABILITY

APPLICATION INTEROPERABILITY

MAINFRAME CONNECTIVITY

INTEGRATED WEB SERVICES FOR J2EE

DYNAMIC NON-STOP REDEPLOYMENT

RAPID ROUND-TRIP J2EE DEVELOPMENT

MASSIVELY INCREASED DEVELOPER PRODUCTIVITY

CLUSTERING FOR LOAD BALANCING AND FAULT TOLERANCE

INTEGRATED DEVELOPMENT TOOLKIT FOR FASTER DEVELOPMENT TIME

TOTAL MANAGEMENT WITH LATEST JMX TECHNOLOGY

PATENTED DISTRIBUTED OBJECT INFRASTRUCTURE

ENTERPRISE MESSAGING

WIDE PLATFORM AND DATABASE SUPPORT

KEY COMPONENT OF IONA'S E-BUSINESS PLATFORM

Download an evaluation copy of our new iPortal Application Server 3.0 at
www.iona.com/devcenter

Java Compatible™
Enterprise Edition

Technologies PCL, IONA e-Business Platform, and IONA iPortal Application Server
rademarks of IONA Technologies PLC. All other product, service, and company names
rademarks of their respective owners.

Total Business Integration™

The two major groups mapping the human genome–Celera and The Whitehead Institute–both independently selected Object Design as the best solution to untangle their complex data management requirements.

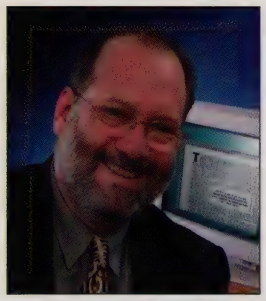
Our products are designed to conquer grueling data management challenges, such as

Object Design

highly distributed environments, massive data sources with complex relationships, extreme performance and unlimited scalability. That's why customers like Amazon, Delta Airlines, Intel, Merrill Lynch, Sony, MCI, Motorola, AT&T Wireless, BT, and Ericsson choose Object Design for their mission-critical applications.

www.objectdesign.com

Services with a Smile



by Steve **GILLMOR**

Web services were the talk of the town at JavaOne this year. Sun's vibrant developer showcase is a kind of instant city—an annual Woodstock celebrating the depth and breadth of the Java revolution. Attendance was down compared to last year's explosive surge—the conference business has seen the same dot-shakeout felt elsewhere—but the frantic rush of a year earlier has been replaced by a mature sense of confidence.

"That's the main benefit that you get from a standard—no surprises." said Sun's Simon Phipps. Kicking off *Java Pro*'s application server roundtable, Phipps said he saw J2EE and the application-server space as an area of exponentially growing opportunity. "Because we have a community-controlled standard in J2EE, we're seeing the ability for us all as vendors to navigate that opportunity space at linear cost rather than constantly being involved in changes that we weren't expecting because the technology is under proprietary control."

> "That's the main benefit you get from a standard—no surprises."
>
> –Simon Phipps

"A year ago, the talk was about licenses and brands. IBM was talking SOAP and XML—Sun was not. Now Sun is knee-deep in Web services," said Peter Kacandes, senior product manager for Java XML APIs. "Our goal is to provide a core set of Java technologies for various XML standard protocols and put them into a pack—called the JAX Pack—so developers can get them all together." J2EE 1.4 will bake the XML APIs directly into the platform.

"We've seen a huge pivot point in the progression of technology as we've gone from business-to-consumer to business-to-business," says Phipps about the move to Web services. "Business-to-business is about the connection of empowered processes, not the capture of consumers. In most businesses, it's the processes that spend the money, not the people. J2EE is the best option for navigating in that space because it provides a sweet point of stability in the technology standard with innovation and progression of new ideas."

Microsoft sees things differently. "The Java vendors and the J2EE vendors would see Web services as a threat," Paul Flessner, senior vice president in the .Net Enterprise Server Division, told us at Microsoft's TechEd developer conference the week after JavaOne. "They're reluctantly embracing it, if they are. They want to sell you on J2EE. It's 'us against Microsoft'—that was the way they positioned it in the market. 'If we all do J2EE, this is all going to work great.' And the reality is Web services don't discriminate—they're agnostic—and that takes away the advantage in the market that they thought they were going to achieve."

Microsoft sees itself as way ahead in Web services. "No one except for Microsoft has commercially supported code," said Barry Goffe, group manager of .Net Developer and Enterprise Communications. He dismissed the raft of Web service announcements coming out of JavaOne as science projects or beta software. With Beta 2 of Visual Studio.Net shipped at TechEd and a Hailstorm SDK coming at October's Professional Developers Conference, Redmond sees a familiar sense of momentum going its way.

"We'd love to see that buzz again, where people just get really excited about making it easy for developers," said Flessner. "We lost that for a while against Java—the market ganged up on us, we got caught in the court, and even though we had faster technology than the competitors, we couldn't use it. But Java and J2EE and the fragility of that environment, the lack of performance that they've worked so hard to get—it's making them vulnerable. And Web services is a more resilient message because it's what the customers really ask for."

Macromedia CTO Jeremy Allaire agrees—with both points of view. "J2EE has only become a standard that's getting some broad adoption in the last year, and .Net hasn't yet shipped. Microsoft [wants] to be the primary provider of the major horizontal APIs that developers and companies need to build applications, just like they did with Windows, but in the networked world. It's a very logical approach, and they'll be successful with it. I don't think they'll be the *only* company doing that." JP

Steve Gillmor, Editor in Chief
sgillmor@fawcette.com

TO THOSE NOT INTERESTED IN RAPID WEB SERVICES DEVELOPMENT

IT'S BEEN NICE KNOWING YOU

AND WE WISH YOU LUCK IN YOUR NEXT CAREER

Only one Web Services environment satisfies your need for speed.

SilverStream eXtend™ Workbench is the only complete integrated environment for the rapid discovery, creation, assembly, consumption and deployment of Web Services. Regardless of whether you're building Web Services from existing components (EJB or Java), from legacy apps (CICS, JMS or 3270) or creating them from scratch, the standards-based eXtend Workbench makes your job easier and faster by automating routine and repetitive tasks.

Now, see what we mean by fast. Download your FREE copy of the SilverStream eXtend™ Workbench at http://www.silverstream.com/extend.
And open your eyes to what's next.

SilverStream eXtend™

www.silverstream.com/extend

©2001 SilverStream Software, Inc. SilverStream eXtend is a trademark of SilverStream Software Inc. All other trademarks are property of their respective owners.

JAVAPro

GROUP PUBLISHER, ENTERPRISE GROUP, Jeannine Barnard

EDITORIAL
EDITORIAL DIRECTOR, Sean Gallagher
EDITOR IN CHIEF, Steve Gillmor
EXECUTIVE EDITOR, Kay Keppler
MANAGING EDITOR, Janaya Reitz
ASSISTANT EDITOR, Charles Ferrara

CONTRIBUTORS
James W. Cooper, Claude Duguay, Rick Grehan
Andy Patrizio, Daniel F. Savarese, Peter Varhol

ART & PRODUCTION
VICE PRESIDENT, ART & PRODUCTION, Michael Hollister
SENIOR ART DIRECTOR, Bruce Gardner
PRODUCTION DIRECTOR, Christine Marquez
ART DIRECTOR, Ginger Cross
ASSOCIATE PRODUCTION MANAGER, Kathleen Sweeney
SENIOR DESIGNER, Kim Le
SENIOR PRODUCER, Sasha Owen
ASSOCIATE WEB PRODUCER, Shane Lee
ASSOCIATE WEB PRODUCER, Mary Wahlen
ASSOCIATE PRODUCER, Mona Bedi
WEB ASSOCIATE, Hao Luu

ADVERTISING SALES
ADVERTISING DIRECTOR, Shana Deane
INSIDE SALES MANAGER, David Seymour
EAST COAST FTP ADVERTISING NETWORK DIRECTOR, Kevin Downey
EASTERN REGIONAL SALES MANAGER, Brian Levine
SOUTHERN REGIONAL SALES MANAGER, Chris Kourtoglou
NORTHWESTERN REGIONAL SALES MANAGER, Janice Mahler
PRODUCTION COORDINATOR, Tami Stoeckle

CIRCULATION
CIRCULATION MANAGER, Meg Clark
ASSOCIATE CIRCULATION DIRECTOR, David Turbert
CIRCULATION ASSOCIATE, Gerry Guzman
CIRCULATION ASSOCIATE, Joyce Hoelzle

MARKETING
SENIOR VP MARKETING, Rob Tobias
MARKETING COMMUNICATIONS MANAGER, Libby Dickinson
MARKETING DESIGNER, Margaret Horoszko
MARKETING PRODUCTION COORDINATOR, Ray Long
MARKETING SPECIALIST, Susan Ogren
DESIGNER, Helga Taylor

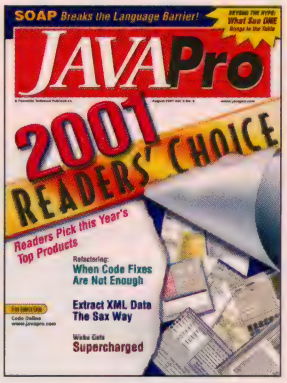
CONFERENCES
VICE PRESIDENT, Tena Carter
EDITOR, WEB CONFERENCES, Robert Scoble
EDITORIAL DIRECTOR, Lee Thé
ASSOCIATE EDITOR, Maryam Ghaemmaghami
CONFERENCE MARKETING MANAGER, Jennifer Brucker
CONFERENCE MARKETING DIRECTOR, Thea Schwartz
CONFERENCE MARKETING PLANNER, Nicole Simonich
CONFERENCE MARKETING PLANNER, Shannon Sullivan
CONFERENCE PLANNER, Staci Matlack
MARKETING EDITORIAL ASSISTANT, Sarah Lemonds
WESTERN REGIONAL SALES MANAGER, WEB CONFERENCES, Cedric Fellows
CONFERENCE SALES PLANNER, WINDOWS GROUP, Heidi Corcoran
WESTERN REGIONAL SALES MANAGER, WINDOWS GROUP, Tanya Cecchin
CONFERENCE OPERATIONS PLANNER, Leslie Hasvold
CONFERENCE ASSOCIATE WEB PRODUCER, Mary Whalen
EASTERN REGIONAL SALES MANAGER, WEB CONFERENCES, Larry Westfall
SPONSORSHIP COORDINATOR, WEB, Jessica Bartlett

CUSTOMER SERVICE
CUSTOMER SERVICE REPRESENTATIVE, Melanie Meinert
CUSTOMER SERVICE REPRESENTATIVE, Laurie Groenewold
CUSTOMER SERVICE REPRESENTATIVE, Nicholas Cadigan

OPERATIONS
CHIEF FINANCIAL OFFICER, John Sutton
DIRECTOR OF HUMAN RESOURCES AND FACILITIES, Philip Ziman
SYSTEM ADMINISTRATOR, Tin Cao
INFORMATION SYSTEM SPECIALIST, Todd Gallagher
ACCOUNTING OPERATIONS MANAGER, Linda Flowers
DIRECTOR OF FINANCE, Darlyn Philips
A/P ACCOUNTANT, Elena Ostrovsky
PAYROLL ACCOUNTANT, Richard Feierstein
PAYROLL SPECIALIST, Linda Darmas
STAFF ACCOUNTANT, Nancy Burgoa
STAFF ACCOUNTANT, Iain Neillands
CASH MANAGEMENT ACCOUNTANT, Betty Wu
COLLECTIONS ANALYST/CASH MANAGEMENT ACCOUNTANT, Lina Yonemitsu
FINANCIAL ANALYST, Neema Buruku
HR MANAGER, Pamela Davis
OFFICE OPERATIONS MANAGER, Nancy Wilson
FACILITIES COORDINATOR, Jamie Po'oi
RECEPTIONIST, Wendy Gray

FAWCETTE TECHNICAL PUBLICATIONS
PRESIDENT, James E. Fawcette
CHIEF OPERATIONS OFFICER, Patricia Cameron-Loyd
VICE PRESIDENT, PUBLISHING, Jeff Miller
VICE PRESIDENT, CHIEF INFORMATION OFFICER, Aaron Weule
VICE PRESIDENT, INTERNET PRODUCTS, Matt Carter
PUBLISHER, FTP ADVERTISING NETWORK, Paul Spyksma
GROUP PUBLISHER, Lynne Matthes
CORPORATE COUNSEL, Wilson, Sonsini, Goodrich & Rosati

Letters to the Editor



Letters to Java Pro *are welcome; they may be edited for form, fit, and style. Please send them to Letters to the Editor,* Java Pro, *c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax them to 650-853-0230; or e-mail them to java-pro@fawcette.com.*

Making More for Less

The salary survey in the July issue finally confirmed it: Technicians and programmers with MBA titles are actually making *less* money than those with BS degrees and are making about the same as those people who have masters degrees in history, music, and so on. Crazy? Believe it.

I've just removed the "MBA" designation from my resume.

M. Simms

Sticky SOAP Situation

I found your June 2001 column "SOAP Your Windows" very interesting, but I am kind of stuck when I try to execute

```
java org.apache.soap.server.
   ServiceManagerClient
   http://localhost:8080/soap/
   servlet/rpcrouter deploy dd.xml
```

This is the answer I am getting:

```
H:\soap-2_1\samples\
   addressbook>java
   org.apache.soap.server.
   ServiceManagerClient
   http:\\localhost:8080\soap\
   servlet\rpcrouter deploy dd.xml

Exception in thread "main"
   java.lang.NoClassDefFoundError:
   org/w3c/dom/Node at
   org.apache.soap.server.
   ServiceManagerClient.<init>
   (ServiceManagerClient.java:79)
   at org.apache.soap.server.
   ServiceManagerClient.main
   (ServiceManagerClient.java:214)
```

After investigation, org.w3c.dom.Node is in the xerces.jar and I have it on my classpath:

```
classpath=h:\MyJavaClass;.;h:\jdk1.3\
   jre\lib\rt.jar;h:\jdk1.3\src.jar;
   h:\soap2_1\lib\
   soap.jar;h:jakarta-
   tomcat\lib\servlet.jar;h:\xerces-
   1_3_1\xerces.jar;
```

This is line 214 of ServiceManagerClient:

```
ServiceManagerClient smc =
   new ServiceManagerClient (
   new URL (args[base]));
   if (base == 2) {
     smc.setUserName (userName);
     smc.setPassword (password);
```

and line 79 is in bold:

```
   public class
     ServiceManagerClient {
     URL routerURL;
     Vector params = new Vector ();
     Call call = new Call ();
     String userName;
     String password;
```

Do you have any idea what is wrong? I thought that if I get to the stage where I need to debug Apache code, there should be something wrong somewhere else.

Michel Petit

James Cooper Responds: It is very important that xerces.jar be *first* in your classpath because some of the other XML methods are also in other libraries. This is probably your problem. Good luck.

InstallAnywhere Enterprise Edition 4

Cross-platform setup without the hassle

by Michiel de Bruijn

When the first Java-based installers were released some years ago, user-friendly cross-platform installers were obviously a cool idea, but their lack of features meant that they were not quite ready to compete with the established platform-specific products. That functionality gap has been pretty much closed now. Although support for some technologies, such as the Microsoft Windows Installer, is still absent for obvious reasons, the current generation of Java-based installers can match their less-portable counterparts almost feature for feature.

InstallAnywhere version 4 is a case in point. It sports a completely redesigned designer interface, making creation of your installation packages even easier than before. Provided you don't need any advanced features, it takes just four steps to package your project, bundle a Java Virtual Machine (JVM), and generate all the files needed by your end users.

Of course, few projects will be as simple as that, so you'll probably spend most of your time using the advanced view, which allows you to customize every step of the setup process, including install sets, billboards, and pre- and post-install actions (see Figure 1).

You can add functionality by using your own Java code and the custom code application programming interface (API). Whereas these extensions essentially were limited to simple noninteractive actions in previous releases, the version 4 API allows for more extensive access to the installer state and user interface.

Quick FACTS New features and enhancements make InstallAnywhere 4 an even better tool to create cross-platform setup programs for both desktop and server applications.

Targeting the Server

In addition to enhanced support for "magic folders," which allow you to install files automatically to locations that vary by operating system (such as the "common files" or "desktop" folders), InstallAnywhere 4 also offers more truly platform-specific features. For example, you can now easily install native system services on Windows NT/2000, as well as integrate with the RedHat Package Manager on Linux.

These enhancements are most useful for developers of server-side software. To further please this audience, InstallAnywhere now also supports unattended and console-based operations. The latter are completely text-based, so they can run on machines without graphical user interface (GUI) capabilities.

Not all features that are available in GUI-based installations have a console equivalent yet, but you can still create a functional text-only setup package that can run a custom install, complete with package selection and all. On the other end of the spectrum, the end-user interface for InstallAnywhere-created packages has been revamped as well. The new GUI looks more streamlined and is more customizable than in previous versions.

The capability to run installation packages directly from a Web page has also been markedly improved. The Web applet, which figures out which platform the end-user is running and directs him or her to the right location based on that, is more verbose and less error-prone.

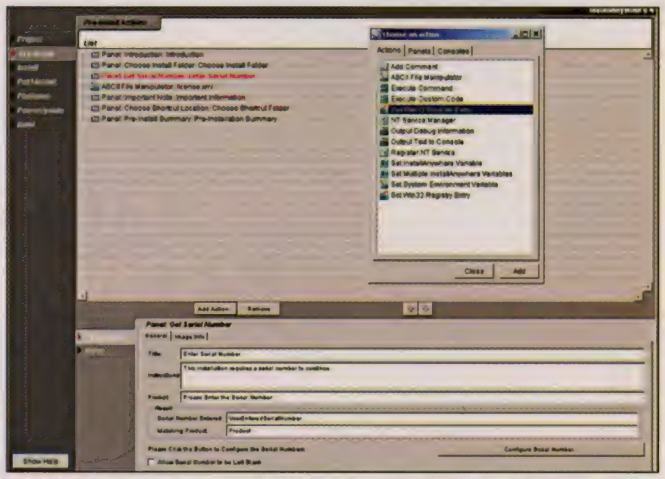


Figure 1 | Getting In on the Action InstallAnywhere's action editor makes it easy to specify custom steps for your installation, including common platform-specific operations.

Installers without Borders

InstallAnywhere 4 is good at adapting to new environments, both from a platform as well as an international point of view. The product comes prepackaged with recent Java virtual machines (JVMs) for the most popular environments: Windows, MacOS, Solaris, and Linux.

Updates can be downloaded from Zero G in the form of "VM packs." If you need to target unsupported platforms or just want to use an alternative JVM, you can easily create these packs yourself.

In addition to being well versed in the different Java flavors, InstallAnywhere also comes prelocalized in 29 languages.

You could find a few things not to like about the product, but you'd have to try hard. For example, some people might be offended by the 1.6 MB taken up by a default InstallAnywhere installation binary without a JVM.

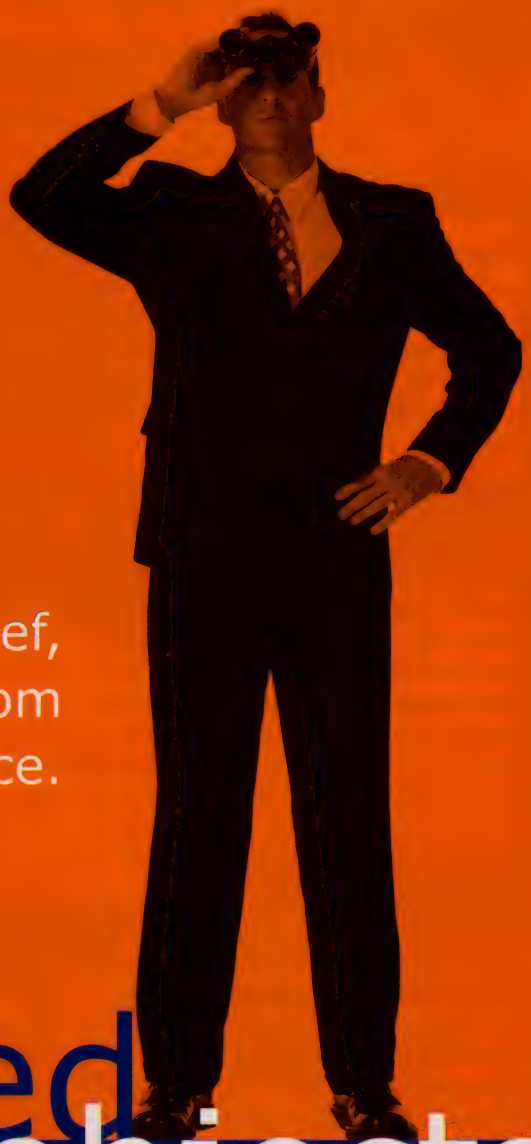
But considering the features offered, as well as the similar (if not larger) requirements of the platform-specific competition, some disk space is a small price to pay.

As far as cross-platform installers go, InstallAnywhere 4 is one of the most advanced offerings out there. If you haven't evaluated the product recently, take a look at the 30-day trial available from the Zero G Web site.

Zero G Software, Inc.
514 Bryant Street
San Francisco, CA 94107
PHONE: (415) 512-7771
WEB: www.zerog.com
PRICE: $2,495
(Standard Edition: $995)

Michiel de Bruijn lives in Rotterdam, The Netherlands, and spends his days as a networking and development specialist for an international media group. Reach him at **mdb@x42.net**.

Ericsson | Echelon | Perkin Elmer | IntelliCorp | KLA-Tencor | BuildNet | Evolve | Lucent | Siemens | US West



Contrary to popular belief, they do not come from outer space.

unidentified flying objects

If you're integrating object persistence into your latest device or application, you won't have to wonder, or worry, or go where no one has gone before. FastObjects™ by Poet is a robust, reliable, high-performance persistence engine that's already been licensed by hundreds of the world's leading manufacturers and developers. Complex data structures? Real-time processing requirements? Familiar territory. FastObjects by Poet. Call us at 1-800-950-8845, or visit our web site at www.poet.com

FAST( )BJECTS™

www.poet.com

In Brief

Libraries • Authoring and Deployment Tools • Database and Middleware Tools

Libraries

Expresso Framwork 3.0

A library of Java servlet components for providing services to Web-based client applications, Expresso Framework 3.0 is designed to allow developers to build entire systems. The program offers secure, high-performance database access via object-to-relational mapping, background job queuing and handling with dynamic invocation of custom server-side objects, and sophisticated event notifications and logging. Its components include security, access to databases, e-mail connectivity, job control, health check, logging, event notification, database objects, download, configuration values, and access to other server-side components. The new features in version 3.0 include data encryption, Apache Log4j and Expresso XML integration, a single error-handling controller, a JSP tag library, and database maintenance capabilities. Free.

JCorporate
Phone: 242-351-4773
E-mail: info@jcorporate.com
Web: www.javacorporate.com

TopLink for WebLogic 3.5 Foundation Library

A superset of the TopLink for Java Foundation Library, with which it shares capabilities, TopLink for WebLogic 3.5 Foundation Library is tightly integrated with container-managed persistent (CMP) EJBs in WebLogic. Starts at $2,000.

WebGain, Inc.
Phone: 877-932-4246; 408-517-3700
E-mail: service@webgain.com
Web: www.webgain.com

Authoring and Deployment Tools

Headway reView 2.5

This source code visualization tool for Java and C++ developers parses class files and source code to reverse engineer a visual representation of the composition and the dependencies of the application. The resulting diagrams, called hiGraphs, use UML notation to provide intuitive, interactive visual environments that developers can use to gain an understanding of how their applications are—or should be—structured. Contact for pricing.

Headway Software, Inc.
Phone: 617-718-0272
E-mail: info@headwaysoftware.com
Web: www.headwaysoftware.com

INSISO

A virtual Internet desktop platform for running Java applications and applets in Web browsers, INSISO is designed for ISPs, ASP pages, or portals that want to provide scalable application services through one or more channels (PC, mobile phone, PDA). It has a standard desktop interface, and runs inside users' Web browsers. The product is equipped with a start menu, a taskbar, a file browser, and a printing service. Its file system and messaging service provide common storage and workgroup functionality for user data and applications. The platform's capabilities include file storage, folder and document sharing, e-mail, calendar, bookmarks, instant messenger, and a text editor. Contact for pricing.

objectBOX
Phone: +47-55-98-1280
Web: www.javabee.com

WebGain StructureBuilder 4.5

A complete UML visual design, development, and deployment tool for J2EE-compliant enterprise applications, WebGain StructureBuilder 4.5 integrates Java and pure UML classes in the same diagrams, affording a development environment for the entire application life cycle. It is available in three editions, with increasing functionalities: Standard, Ex-

Java and J2EE tools for the Solaris Operating Environment

Rational Software has teamed with Sun Microsystems to deliver new Java and Java 2 Enterprise Edition (J2EE) tools for the Solaris Operating Environment. The tools are designed to work with Sun's existing toolsets, including Forte for Java 3, which was announced at JavaOne. Rational and Sun promise that the combination of their development tools will provide a single environment for creating J2SE, J2EE, and the newly introduced Java 2 Micro Edition (J2ME) platforms.

The deal focuses on three areas: support for the Solaris 8 environment, with Rational's full suite of tools on Solaris, including Rational Rose; the integration of modeling tools, Rational ClearCase and ClearQuest, which handle change management and bug fixing; and the integration of Rational Test and Rational RealTime, which are used for testing.

A number of Rational products, including Rose and ClearCase, will be integrated with the Forte for Java IDE. The integration of Forte with Rational Rose RealTime and Rational Quality Architect RealTime provides the first modeling and testing suite for J2ME development.

Finally, Sun and Rational's service groups will work together to support developers using the two products, and they will combine source code from the J2EE Patterns Catalog with Rational design templates to help accelerate development.

No deadline was given for when the integration and new support options will be available, but all Rational products are available now at www.rational.com.

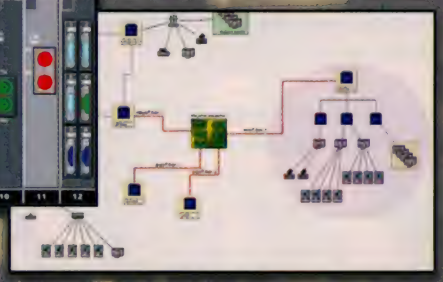
Visualizing a Managed World™

LOOX Software's products provide a wide range of capabilities that are tailored to match our customers' specific development needs.

JLOOX™ 2.0

The Award Winning Graphics Toolkit for Java2D™ Developers

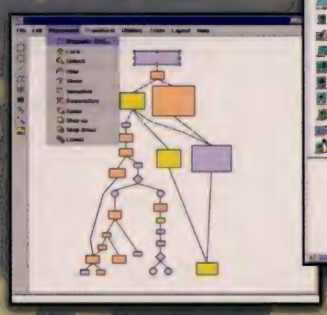
Element Management System



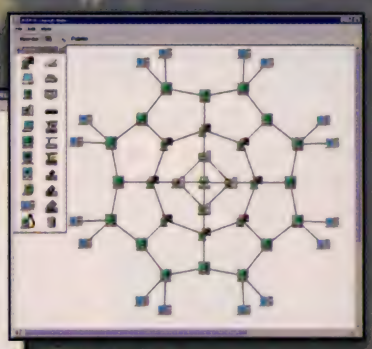
Network Management System

JLOOX Layout™ 2.0

The Development Tool for the Automatic Layout and Visualization of Complex Topologies



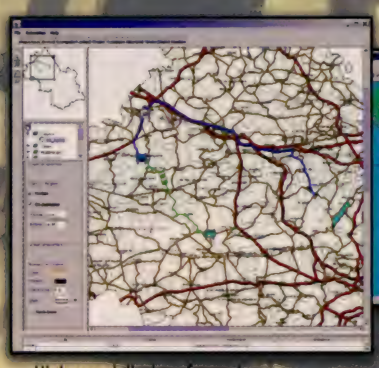
Hierarchical Layout



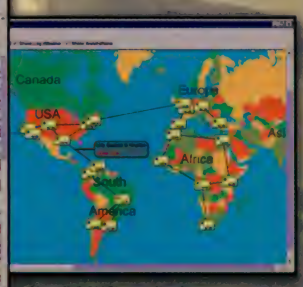
Spring Layout

JLOOXGis™ 2.0

High-Performance Map Rendering Components for Location Based Visualization Applications



High speed display of complex maps



Use JLOOXGis to display distances and animated graphical objects

Download a 30 day trial from www.LOOX.com

Applications created with LOOX Software's products have no runtime, royalty or deployment fees.

USA

LOOX Software Inc.
1860 El Camino Real, Suite 306
Burlingame, CA 94010

Tel: 650.552.9310 or 1.800.684.LOOX
Fax: 650.552.9307
Email: info@loox.com

LOOX SOFTWARE
A subsidiary of eNGENUITY Technologies

EUROPE

LOOX Software S.A.
1, rue Eugène Varlin
92240 Malakoff - FRANCE

Tel: +33.1.49.65.32.70
Fax: +33.1.49.65.49.72
Email: info-europe@loox.com

pert, and Enterprise. The new features in version 4.5 include enhanced UML support for state and activity diagrams, integration with WebGain Studio 4.5, a new user interface, improved project navigation and management, and a new layout engine to assist in initial class diagram layout. Free Standard Edition; $995 Expert Edition; $1,995 Enterprise Edition.

WebGain, Inc.
Phone: 877-932-4246; 408-517-3700
E-mail: service@webgain.com
Web: www.webgain.com

Database and Middleware Tools

Allora for Java 1.5

This database-focused DOM/SAX implementation is designed to help maintain an XML-consistent interface across all relational databases by allowing both read and write database integration. Developers can use Sun standard JAXP interfaces to build W3C DOM objects, transform them as necessary, update node values, and apply the XML data changes back into the underlying DBMS. Allora for Java 1.5 supports JDK 1.2 versions and later. It is available in Java and native Windows versions and is compatible with all major Web application development platforms, including BEA Weblogic, IBM WebSphere, iPlanet, SilverStream, and others. Version 1.5 offers data-binding capabilities and message queue support. Contact for pricing.

HiT Software
Phone: 408-345-4001
E-mail: info@hitsw.com
Web: www.hit.com

db4o 1.0

A Java object database optimized for wireless devices, db4o 1.0 is designed for high Java language integration, omitting driver overhead, and maximizing speed. Object persistence is not dependent upon making changes to application classes. A query-by-example feature minimizes resource consumption and improves performance, and automatic object recognition permits zero-admin deployment. The program also features a small footprint that saves device memory and allows online downloads. Starts at $1,000.

db4o
Phone: 888-328-2742; 703-802-8210
E-mail: info@db40.com
Web: www.db4o.com

e-Content Management Platform 3.3

This Web-based application with a Java/J2EE architecture is designed to centralize, search, control, classify (via metadata), version, and track electronic business files regardless of their formats. The system is open and extensible, capable of leveraging resources such as full text search systems, databases, application servers, XML index servers, file systems, and legacy/custom applications. It is comprised of an engine, a developer suite, and a Web client. System customizations are possible through API, JSP, servlets, applets, and policies. Contact for pricing.

Quickstream Software, Inc.
Phone: 719-520-1878
E-mail: info@quickstream.com
Web: www.quickstream.com

Kada Mobile Developer Studio Alliance Program

For developers who want to put an Oracle or Sybase database on a Palm Pilot, Kada Systems has begun the Kada Mobile Developer Studio Alliance Program, which is designed to support the development of Java-based database applications on handheld devices. Java IDE vendors, such as WebGain and Borland, can license the Kada Mobile Platform and Kada Mobile Developer Studio tools. The Mobile Developer Studio is a set of tools for writing, testing, tuning, and optimizing Java applications for handheld devices. Kada currently supports Palm Computing's PalmOS and Microsoft's Pocket PC platforms.

At the same time, Kada has teamed with iAnywhere, a subsidiary of Sybase, for the purpose of simplifying cross-platform development between enterprise and mobile databases. iAnywhere develops the m-Business platform, which lets workers access their corporate data via a handheld device.

Obviously, the mobile implementations of Oracle and Sybase are significantly scaled down from their enterprise equivalents, but the reason for porting them to mobile devices is to preserve data integrity. "This allows you to work in the environment that you are familiar with from the office," said James Acquaviva, CEO of Kada Systems. "This lets you take your data and a subset of your applications on the road with you, and you can take different applications with you when you need them."

Kada's tools are available for a free trial download at www.kadasystems.com.

WebGain Studio Professional and Standard Edition 4.5

WebGain, Inc. has released WebGain Studio Professional and Standard Edition 4.5, an update to its enterprise Java development suite. WebGain Studio consists of the VisualCafé Java IDE, StructureBuilder, a modeling tool (TopLink), which builds connections between applications and data sources, and Quality Analyzer, the analysis tool.

Most of the changes to WebGain Studio 4.5 are enhancements to the existing technologies, not major new additions. This is the first release to fully support Java 2 Enterprise Edition 1.2, including Enterprise JavaBeans 1.1, and it has support for iPlanet Enterprise servers from the SunINetscape alliance. This is also the first release to include TopLink and Quality Analyzer.

Other changes in WebGain Studio 4.5 include improved JavaServer Pages (JSP) debugging, support for Extensible Markup Language Metadata Interchange (XMI), which lets Rational Rose users import their models into WebGain, and Unified Modeling Language (UML) activity and state diagrams.

"WebGain Studio 4.5 is the next step from an IDE for Java, because it leverages technologies that make the development process a lot easier for developers, so they don't have to spend a lot of time working on application development programming nuances that are automated in studio," said Matt Elson, senior director of product marketing at WebGain.

Version 4.5 of WebGain Studio Professional and Standard Edition are available for trial download from WebGain's Web site: www.webgain.com.

Build your Web Services in 28 Minutes!
1. Publish existing EJB and CORBA enterprise components as Web Services.
5 minutes
2. Create WSDL directly from the deployed logic.
1 minute
3. Generate Java and VB clients that talk to the new SOAP Web Services.
2 minutes
4. Generate XSLT mapping to map between existing XML interfaces and Web service invocations.
20 minutes

Cape Clear makes building and linking Web software *easier and faster*

PLATFORM

CapeConnect™ Web Services Platform lets you dynamically create XML SOAP based Web Services systems from new and existing Java, EJB and J2EE components.

TOOLS

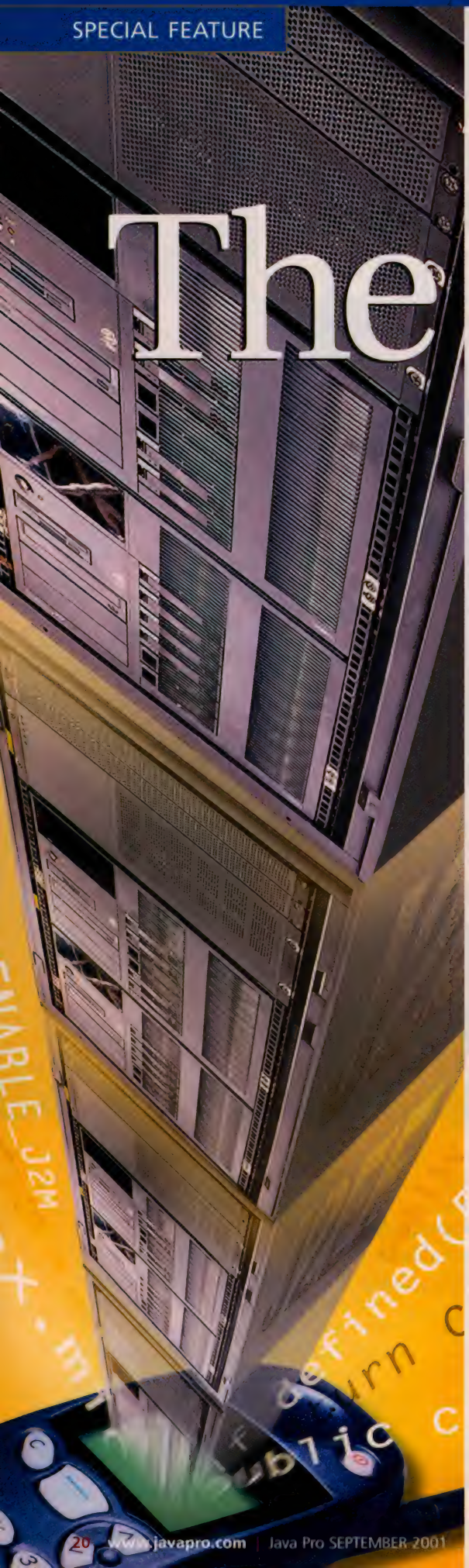
CapeStudio™ enables rapid development of Web Services clients in VB and Java and provides a simple graphical environment for the generation of XML and SOAP transformations in XSLT.

Cape Clear invites you to evaluate our products, read whitepapers and download demo's at http://www.capeclear.com/cc21

✆ 1866 CAPE226 | WWW.CAPECLEAR.COM

CAPE CLEAR™
Automating the Business Internet

© Cape Clear Software Inc. Cape Clear, CapeStudio and CapeConnect are trademarks of Cape Clear Software Inc. All other product, service and company names are trademarks or registered trademarks of their respective owners.



The Great Migration

J2ME presents portability problems not found in other parts of the Java platform; overcoming them is critical for consumer satisfaction

by Daniel F. Savarese

Java's original mantra was "write once run everywhere," but over time it has evolved into "one size doesn't fit all." Even though the early days saw many platform-specific hiccups, it was largely possible for most projects to maintain a single source base that would work for all platforms—that is, if you ignored the difficulties in creating debug and test versions of Java code caused by the lack of conditional compilation support. Each revision of the Java platform introduced cross-JDK portability problems, but many projects transitioned smoothly to the latest JDK without having to support multiple JDKs in great part because of Java's entrenchment on the server. Had Java been a success on the desktop, the story would have been far different, as evidenced by the major portability and code maintenance dilemmas that plagued developers who bought into applets.

With millions of consumer devices capable of running Java applications entering the mass market, writing portable Java code is an issue no Java developer can escape. Abandoning support for the last incarnation of Java will no longer be an option. Consumers won't stand for not being able to use the latest versions of their favorite applications on their PDAs and mobile phones just because a new JDK was released. Migrating existing code to the new world created by Java 2 Micro Edition (J2ME) will be a challenge, as will the maintenance of J2ME code across the spectrum of J2ME-capable devices.

J2ME had little substance for developers from 1999 through much of 2000. The only Java development possible for small devices was through a set of nonstandard third-party offerings and eventually the experimental KVM distribution for Palm OS. PersonalJava was too bloated to be of much use to anyone, but most of all, no devices on the market could run Java. If there was a frustration among budding J2ME developers, it wasn't the absence of the market—we knew it would emerge eventually. The frustration was with the

Daniel F. Savarese is an independent software developer and technology advisor. He has been the founder of ORO, Inc., a senior scientist at Caltech's Center for Advanced Computing Research, and vice president of software development at WebOS. Daniel is the original author of the Jakarta-ORO text processing packages and the NetComponents network protocol library. He is also a coauthor of *How to Build a Beowulf* and is working on several computer science book projects. He can be reached at **dfs@savarese.org**.

slothful pace at which Sun trickled information about what was going on with J2ME. PersonalJava ran on WindowsCE, but everyone wanted to know when they could develop for Palm OS, mobile phones, and pagers. For a long time, the relationship between PersonalJava and J2ME wasn't even clear. It's taken a lot of waiting and squandered time pursuing nonstandard development paths, but the veil has lifted and the problem has become one of sorting out the jumble of specifications flowing out of the Java Community Process.

It is clear now that J2ME will have a series of layers presenting portability problems not found in other parts of the Java platform. Whereas J2EE is a direct superset of J2SE, able to run every J2SE program, J2ME has diverging nooks and crannies. Every J2ME program will not run on J2SE, and within J2ME, programs written for certain classes of devices will not run on others. The basic composition of J2ME at the time of this writing is depicted in Figure 1 and is explained in the rest of this article. It surely will get more complicated with time.

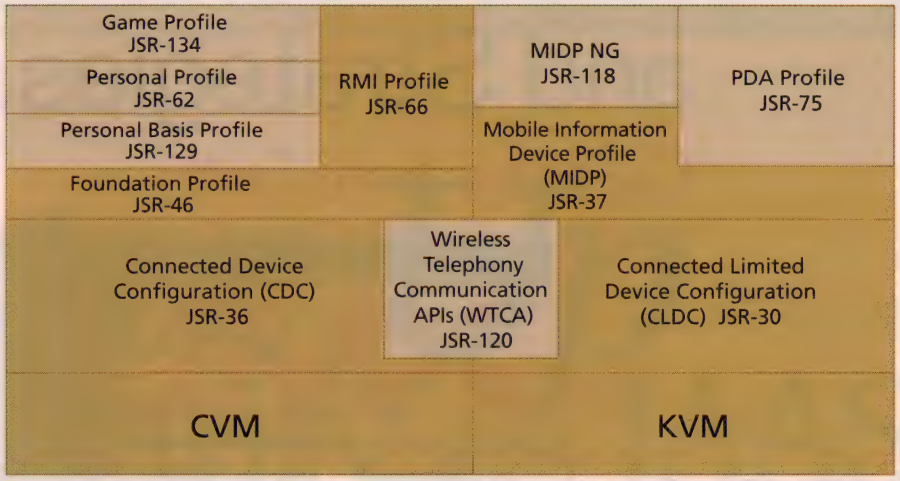


Figure 1 | The Many Faces of J2ME J2ME is not a single entity. It is divided into two major configurations and multiple profiles, complicating attempts to write cross-edition portable Java. The shaded elements have at least reached a public review stage with a proposed final draft. Remaining elements are still under development by their working groups.

Configurations

The J2ME architecture is designed with flexibility and future growth in mind. The basic building block of a J2ME runtime environment is a *configuration*. A configuration consists of a virtual machine that implements some portion of the Java language and virtual machine specifications, and a minimal supporting set of class libraries and APIs. It may seem redundant to say libraries and APIs, but APIs include things like the Java Native Interface (JNI), which are not implemented by a class library or defined by the JLS. Additional features for specific types of devices can be built on top of a configuration, but each configuration defines a base level of functionality for a class of devices. Therefore your first portability target is the configuration.

> With millions of consumer devices capable of running Java applications entering the mass market, writing portable Java code is an issue no Java developer can escape

J2ME defines two configurations, the Connected Device Configuration (CDC) and the Connected Limited Device Configuration (CLDC), but more may be defined in the future provided a new class of devices warrants it. CDC requires full support for both the Java Language Specification and Java Virtual Machine Specification. If you layer enough APIs on top of it, you'd wind up with J2SE plus some extra CDC-specific classes. Therefore the main concern when porting J2SE code is one of which core classes you can assume are available. The complete list of classes included in CDC is described in the CDC specification, available from http://java.sun.com/aboutJava/communityprocess/jsr/jsr_036_j2mecd.html. The major gotchas are in the java.io, java.net, and java.security packages.

The classes you will miss the most in a straight-up CDC environment are Socket, ServerSocket, HttpURLConnection, CharArrayReader, CharArrayWriter, StringReader, and StringWriter. Even though CDC is missing TCP socket support, it does include the J2SE UDP DatagramSocket class and the CLDC javax.microedition.io.DatagramConnection interface. URLs referenced by "file:" and "datagram://" protocols will therefore work. Perhaps the most important thing to remember is that CDC and the profiles built on top of it have removed support for almost all J2SE deprecated APIs. If you've got any calls to deprecated methods or classes in your code, you'll have to update them to use the most recent APIs. Java legacy code is no longer an oxymoron.

CDC does not require that JNI, the JVM Debugging Interface (JVMDI), or the JVM Profiling Interface (JVMPI) be implemented. However, if a native method invocation mechanism is provided, it must implement JNI 1.1. If a debugging interface is provided, it must implement JVMDI. And if a profiling interface is provided, it must implement JVMPI. CDC does not define any user interface classes, so forget about using AWT and Swing. User interface classes are defined in additional profiles, as are TCP networking facilities and RMI. For the most part, J2SE programs that stick to the core classes of java.lang, java.io, and java.util won't need any modification for use with CDC. The entire Collections Framework is included in CDC, which takes care of most core programming needs.

The JVM that is provided with the reference implementation of CDC is called the CVM. The C presumably reflects that the virtual machine is targeted at consumer and embedded devices. It has a smaller footprint than the J2SE JVM and is optimized

Continued on page 88.

XML Solutions and Strategies:

Get them here first!

How to Choose an XML Ser
XML
MAGAZINE
.NET=XML
How XML Will Move Microsoft Beyond Windows
Apps to Web Services with SOAP
Serving XML with JavaServer Pages
Expert Voices: .NET's Wascha on XLANG, Hejlsberg on C#
Cocoon Separates Logic from Layout
Microsoft .net

FREE Subscription!

XML Magazine is the only independent publication devoted 100-percent to the latest information about using XML for enterprise-level challenges. Created by the same experts who gave you *VBPJ, Exchange & Outlook* and *Java Pro,* this is the one indespensible XML resource that you can't afford to be without. Published every other month, you'll get practical, hands-on articles, tips and enterprise-level solutions, including:

- Pragmatic techniques for developing with XML
- Cutting-edge analysis of industry trends
- Independent product reviews and new product announcements
- Columns by the industry experts who design the technology
- Complete source code examples

Sign up today for your FREE subscription @ www.xmlmag.com/free

FTP FAWCETTE TECHNICAL PUBLICATIONS

209 Hamilton Avenue, Palo Alto, CA 94301-2500 • Phone: 650.833.7100 • Fax: 650.853.0230

©2000, FTP, Inc. XML is a trademark of MIT. Publisher reserves the right to determine qualifications for free subscriptions.

Jazz Up Java Security with JAAS

The Java Authentication and Authorization Service extends Java's existing security infrastructure

by Tarak MODI

Tarak Modi has been building scalable, high performance, distributed applications for more than six years and has been working with Java exclusively for the past three years. Tarak has a bachelor's degree in electrical engineering, a master's degree in computer engineering, and an MBA concentrating in information systems. He is currently an architect at North Highland, a technology and management consulting company. To find out more about him, please visit his website at http://tmodi.home.att.net or you can e-mail him directly at **tmodi@att.net**.

Security has always been a hot button for software developers. Couple that with Java's built-in elaborate security support and it's no wonder Java became an instant success as soon as it hit the market. Starting from version 1.0 to the recently released version 1.3, Java's security infrastructure has been changed and improved. The most significant set of changes was probably in version 1.2 with the introduction of the access controller and the resulting diminished role (and complexity) of the security manager. A powerful combination of class loaders, security managers, and access controllers gives Java developers enormous support in creating "secure" applications. Now the Java Authentication and Authorization Service extends Java security even further.

Security Shortcomings

Consider an extremely simple Java application (see Listing 1). The selection of the name "UnAuthenticatedClient" soon will become clear. The program checks to see if a file named "ProtectedFile.txt" exists in the current directory. Compile and run the program (see Listing 2).

Everything should work fine. By default, Java applications execute with no security in an open sandbox (or no sandbox). Now let's start closing the sandbox. This time run the program like this:

```
REM Run it
C:\jdk1.3\jre\bin\java -classpath
  C:\dev\Java\javapro\jaas\
  noauth_client.jar -
  Djava.security.manager -
  Djava.security.policy==
  C:\dev\Java\javapro\jaas\
  config\noauth_java.policy
  javapro.jaas.UnAuthenticatedClient
```

We tell the Java virtual machine (JVM) to use a security manager and define the security policy in the file C:\dev\Java\javapro\jaas\config\noauth_java.policy, like this:

```
grant codebase "file:C:/dev/Java/
  javapro/jaas/noauth_client.jar" {
  permission
    java.io.FilePermission
    "ProtectedFile.txt", "read";
};
```

Translated into simple English, the security policy states that Java code in the JAR file, C:/dev/Java/javapro/jaas/noauth_client.jar, is allowed "read" access to the ProtectedFile.txt file in the current directory.

The results from executing the first and second run commands should be the same at this point. Try running the second run command after editing the policy file so that "read" access is no longer permitted:

```
grant codebase "file:C:/dev/
  Java/javapro/jaas/
  noauth_client.jar" {
};
```

This time you should see a nasty exception. Why? We took the permission away. Now go back to the original policy file, which adds the permission back in. Now copy the JAR file in the C:\temp\ directory and run this command:

Listing 1 Starting Simple

```
package javapro.jaas;
import java.io.File;
public class UnAuthenticatedClient
{
  public static void main(String[] args)
  {
    File f = new File("ProtectedFile.txt");
    System.out.print(
      "\nProtectedFile.txt does ");
    if (!f.exists())
      System.out.print("not ");
      System.out.println("exist.");
  }
}
```

This simple program checks to see if a file named "ProtectedFile.txt" exists in the current directory.

Listing 2 Getting Results

```
REM Assume that the JDK is installed in
  C:\jdk1.3
REM Assume that the Java source files are
REM in the directory C:\dev\Java\javapro\jaas

REM Compile
C:\jdk1.3\bin\javac -classpath C:\dev\Java\
  -d ..\..\ UnAuthenticatedClient.java

REM Jar up
C:\jdk1.3\bin\jar cvf noauth_client.jar
  javapro\jaas\UnAuthenticatedClient.class

REM Run it
C:\jdk1.3\jre\bin\java -classpath
  C:\dev\Java\javapro\jaas\noauth_client.jar
  javapro.jaas.UnAuthenticatedClient
```

This is the code we use to compile and run the program in Listing 1.

```
C:\jdk1.3\jre\bin\java -classpath
  C:\temp\noauth_client.jar -
  Djava.security.manager -
  Djava.security.policy==
  C:\dev\Java\javapro\jaas\
  config\noauth_java.policy
  javapro.jaas.UnAuthenticatedClient
```

You will get another exception, because the Java code no longer comes from the JAR C:/dev/Java/javapro/jaas/noauth_client.jar.

Let's make one final modification to the policy file:

```
grant codebase "file:C:/dev/
  Java/javapro/jaas/
  noauth_client.jar" {
  permission
  java.io.FilePermission
  "ProtectedFile.txt", "read";
};

grant signed by tmodi
codebase "file:C:/dev/
  Java/javapro/jaas/
  noauth_client.jar" {
  permission
  java.io.FilePermission
  "ProtectedFile.txt",
  "read,write";
};
```

This policy file still allows the same access as before, but if the JAR file is signed by tmodi, then it also allows "write" access to the file. What this tells us about Java's security is that you can control the actions a Java application can perform based on where the code comes from and who created it (or signed it).

What if you want to allow only "read" access to the code if user "Bob" executes it? As things stand now, you're out of luck. Java security cannot help you, because Java security has no concept of user authentication and authorization (hence the name UnAuthenticatedClient).

Enter the World of JAAS

To address this exact security issue, Sun has introduced the Java Authentication and Authorization Service (JAAS). As is obvious from its name, the JAAS infrastructure can be divided into two main components: an authentication component and an authorization component.

> JAAS allows different authentication models to be plugged in at run time while shielding the client applications from the specifics of the authentication algorithm

The JAAS authentication component provides the ability to determine reliably and securely who is currently executing Java code, regardless of whether the code is running as an application, an applet, a bean, or a servlet. The JAAS authorization component supplements the existing Java 2 security framework by providing the means to restrict the executing Java code from performing sensitive tasks depending not only on its location and signer (as is done in Java 2), but also on who is authenticated. JAAS authentication is performed in a pluggable fashion, which permits Java applications to remain independent from underlying authentication technologies. Therefore, new or updated authentication technologies can be plugged under an application without requiring modifications to the application itself.

Once the user executing the code has been authenticated, the JAAS authorization component works in conjunction with the existing Java 2 access control model to protect access to sensitive resources. Unlike in Java 2, where access control decisions are based solely on code location and code signers (called a CodeSource), in JAAS access control decisions are based on the executing code's CodeSource, as well as on the user (or Subject) running the code.

Using JAAS

Let's see how to use JAAS in your applications. (You can find a theoretical discussion about JAAS on Sun's Web site at http://java.sun.com/security/jaas/doc/

Java Pro Magazine Introduces

NOVEMBER 28-30, 2001

NEW ORLEANS

JAVA™ TECH SUMMIT

Learn to Build Better, Dynamic Web Sites

With 24 sessions spanning two days, Java Tech Summit is the leading "how-to" event for developers and their managers who are building Internet and intranet sites with the Java server-side programming platform and Java-based application server technologies.

Hear Keynotes from Industry Pioneers

Simon Phipps

Sun Microsystems Chief Technology Evangelist, Explores the Past, Present and Future of Enterprise Java

Michael Smith

BEA Principal Evangelist, E-commerce Server Division, Pushes the Envelope with Web Services

1. Build Dynamic Web Sites with JSP
2. Integrate Enterprise Data and Applications with EJB
3. Create Smart Web Services with XML and Java
4. Leverage JSP Architecture and Servlets to Develop Wireless Apps
5. Harness the Core of J2EE Development

Register by October 17, SAVE $200!

800-848-5523 • 650-833-7100

http://conferences.devx.com/javatechsummit

Java Tech Summit is part of Web Builder 2001.

Your entire team can learn how to build better Internet and intranet sites, applications, and services. For more information on the Next Generation Internet conference, go to **http://conferences.devx.com/webbuilder.**

FTP FAWCETTE TECHNICAL PUBLICATIONS

JAVAPro

Thunder Lizard Productions
Conferences with Content

Java is a trademark of Sun Microsystems. Java Pro and Java Tech Summit are trademarks of Fawcette Technical Publications, Inc. All other trademarks are property of their respective owners.

Listing 3 Waiting for a Callback

```
class MyCallbackHandler implements
  CallbackHandler {

  public void handle(Callback[] callbacks)
    throws IOException,
    UnsupportedCallbackException {

    for (int i = 0; i < callbacks.length; i++) {
      if (callbacks[i] instanceof NameCallback) {

        // prompt the user for a username
        NameCallback nc =
          (NameCallback)callbacks[i];

        // ignore the provided defaultName
        System.err.print(nc.getPrompt());
        System.err.flush();
        nc.setName((new BufferedReader(
          new InputStreamReader(
          System.in))).readLine());

      } else if (callbacks[i] instanceof
        PasswordCallback) {

        // prompt the user for sensitive
        // information
        PasswordCallback pc =
          (PasswordCallback)callbacks[i];
        System.err.print(pc.getPrompt());
        System.err.flush();
        pc.setPassword(readPassword(System.in));

      } else {
        throw new UnsupportedCallbackException(
          callbacks[i], "Unrecognized Callback");
      }
    }
  }
}
```

The callback looks at each callback object in the array of callbacks, gathers the username and password from the user, and returns them to the calling login module.

acsac.html.) We'll revisit the unauthenticated client example we started with, but this time we'll use JAAS to add authentication and authorization support. Because it's no longer fair to call this client unauthenticated, I will rename the new class AuthenticatedClient. The complete implementation of this class is available online at www.javapro.com.

JAAS does not specify any particular authentication technology/algorithm. Instead JAAS allows different authentication models to be plugged in at run time while at the same time shielding the client applications from the specifics of the authentication algorithm. Clients always interact with JAAS through an object called the login context. The login context, in turn, communicates with a login module, which is specified at run time and is responsible for implementing and performing the authentication. Because the login context never changes and the login module exposes a standard interface, no code change or recompilation is required to change authentication algorithms. Rather it is simply a matter of specifying a different login module at run time.

These are the steps taken by the authenticated client:

1. Create a new login context.
2. Use the login context to perform the login.
3. Check to see if the file ProtectedFile.txt exists in the current directory.
4. Use the login context to perform the logout.

Now let's take a closer look at each of these steps.

Create a new login context Creating a new login context is easy:

```
// use the configured
// LoginModules for the
// "JAASArticle" entry
LoginContext lc = null;
try {
  lc = new LoginContext(
    "JAASArticle", new
    MyCallbackHandler());
} catch (LoginException le) {
  le.printStackTrace();
  System.exit(-1);
}
```

Basically, I need two pieces of information: the name of the context and a reference to a callback handler. All login contexts require a name (I'll tell you how to know what name to use in just a minute). All login contexts do not, however, require a callback handler. In our case a callback handler is required because the login module that we will use requires a callback handler to get the username and password at run time.

Now back to deciding the name of the login context. Any application that uses JAAS needs to be configured with information about the login module that it (JAAS) should use to perform the authentication during login. One of the ways of specifying this configuration is with a configuration file specified in the java.security.auth.policy system property:

```
java -Djava.security.auth.policy=
  jaas.config ...
```

The name of the configuration file is immaterial. In our case it is called jaas.config and contains:

```
JAASArticle {
  javapro.jaas.LoginModuleImpl
  required
  debug=true
  fileName=passwords;
};
```

Here JAASArticle is the name that identifies the "configuration block" (within the {}) that follows it. The configuration file can have any number of such blocks; the only condition is that each block have a different name. This is the same name that is specified as the name of the login context. The connection between the two (the login context name and the name of the configuration block) should now be obvious: the login context uses this name to find the appropriate "configuration block" within the specified configuration file. The configuration block informs JAAS about the login modules that must be used to perform the authentication.

JBuilder™

www.borland.com/new/jp_jb5.html

When you're out in front, life is exciting. JBuilder™ 5 puts you there on the cutting edge of cross-platform Java™ development. Rapidly create Enterprise JavaBeans™ and Web applications. Deploy to the leading J2EE™ e-business platforms—Borland AppServer,™ WebLogic,® and WebSphere.® Improve team productivity with ClearCase,® CVS and Visual SourceSafe™ integration. Publish and exchange data using XML. Find out for yourself why JBuilder is the #1 Java development environment.

Borland®

Take the Lead.
Get JBuilder.
Call 1-800-252-5547

Mention offer code: 1784

Free Enterprise Studio Seminar Tour
http://www.borland.com/events/seminars/entstudio

Made in Borland® Copyright © 2001 Borland Software Corporation. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other marks are the property of their respective holders. • 12038.4

We will be using the login module that we've already developed. In addition to specifying the fully qualified class name of the login module, any options specific to that login module are also specified here. We have specified two such options because our login module needs the name of the passwords file and an optional "debug" on/off parameter. Here the name of the passwords file is "passwords" and it will be created using the password management utility described later. Each configuration block can have any number of login modules specified in it.

The creation of the login context causes JAAS to load all the login module(s) specified in the configuration block and call the initialize method on each one of these modules.

Use the login context to perform the login Once the login context has been created, we can use it to perform the login by calling the login method on it:

```
try {
  // attempt authentication
    lc.login();
}
catch (LoginException e) {
  System.out.println(
    e.getMessage());
    System.exit(-1);
}
```

Calling the login method on the login context results in the login method on the login module(s) being called. Depending on the result of these calls, JAAS will call the commit() or abort() method on the login module, as well. As far as the login module goes, logging in is a two-step process (similar to the two-phase commit protocol used by transaction managers). The login method implementation of our login module performs a callback on the callback handler that we provided earlier. A callback handler must implement the javax.security.auth.callback.Callback interface, which has one method—handle()—that gets an array of callback objects from the calling login module. Callbacks provide a means for JAAS application clients to be notified of certain events during the authentication process and for providing data to the authentication process. The implementation of our callback handler is shown in Listing 3.

Check to see if the file ProtectedFile.txt exists in the current directory At this point, JAAS has authenticated the client. Now we're interested in performing a check to see if the file ProtectedFile.txt exists in the current directory. Remember, we want only authorized users to be able to do this—whether or not the "file exists" check is allowed depends on who's running the code. That's where JAAS comes in. For JAAS to help us, we can't just create and access the file as before. If we do, then JAAS does not have anywhere to intercept and perform the authorization.

To perform an authorized access, we need to gain access to the Subject object. A Subject object encapsulates an individual or oganization with multiple principal identities, along with a set of private and public credentials. Having clarified that, the next question is how do we gain access to the

Listing 4 New Password

```
C:\jdk1.3\bin\java -classpath
  C:\dev\Java\  javapro.jaas.PwdUtil

Welcome to the Password Management Utility.
Type help at any point for a list of
  supported commands.

cmd>create
cmd>add tmodi a
cmd>add mala 123
cmd>list
tmodi
mala
cmd>saveas passwords
Passwords Saved.
cmd>help
Supported commands: add, change, help, list,
  load, remove, save, saveas, and quit
cmd>quit
Goodbye...
```

You can create a new password file using the password utility.

Listing 5 User Info

```
// prompt for a username and password
// To do so the callback handler must have
// been set in the initialize method.
if (callbackHandler == null)
  throw new LoginException("Error: no
    CallbackHandler available " +
    "to garner authentication information
    from the user");

// Create the callback array
Callback[] callbacks = new Callback[2];
// one callback for the name
callbacks[0] = new NameCallback("username: ");
// and another for the password
callbacks[1] = new PasswordCallback(
  "password: ", false);

try {
  // invoke the handle method on the callback
  // handler passing it the callback object array
  callbackHandler.handle(callbacks);
    // and now extract the username and password.
    username =
      ((NameCallback)callbacks[0]).getName();
    char[] tmpPassword = ((
      PasswordCallback)callbacks[1]).getPassword();
    if (tmpPassword == null) {
      // treat a NULL password as an empty password
      tmpPassword = new char[0];
  }
  password = new char[tmpPassword.length];
  System.arraycopy(tmpPassword, 0, password,
    0, tmpPassword.length);
  ((PasswordCallback)callbacks[
    1]).clearPassword();

}
.
.
.
```

The first step in our login() method is for the method to use the callback object to get the username and password from the client.

The ***Fastest and Most Scalable***
Java Messaging Server

FioranoMQ 5

now adds:

XML Content-based Routing to deliver personalized content in real-time to tens of thousands of concurrent subscribers using XPath Selectors.

A unique and proven high-performance ***File-based Datastore*** together with enterprise ready features including ***LDAP and REALMS integration***, ***SOAP support***, pluggable ***Scalable Connection Management (SCM)*** module and more, makes FioranoMQ the messaging infrastructure of choice for the world's leading financial and telecommunications companies including AT&T Wireless, Morgan Stanley, Motorola and others.

Fiorano

Distributed Computing Infrastructure

Download your ***Free Developer License***
http://www.fiorano.com

Subject object in the client? Simple. By using the getSubject() method on the login context. The Subject object also has a static method called doAs(), which provides the entry point for performing an authorized access. This method takes two parameters: the subject object that will provide the identity of the caller, and a reference to an object that implements the java.security.PrivilegedAction interface. Our implementation of that object goes like this:

```
package javapro.jaas;
import java.io.File;
import
  java.security.PrivilegedAction;
public class AuthorizedAction
  implements PrivilegedAction {
  public Object run() {
    File f = new File(
      "ProtectedFile.txt");
    System.out.print(
      "\nProtectedFile.txt does ");
    if (!f.exists())
      System.out.print("not ");
      System.out.println("exist.");
    return null;
  }
}
```

The run() method implementation is basically the same code that appeared in the unauthenticated client that was not subject to any authorization. For the authorization to take place properly, not only must the regular Java security policy file be set up, but an additional JAAS security policy file must be set up as well. Here is the relevant portion of the Java security policy file:

```
// note: client.jar contains
// AuthenticatedClient.class
grant codebase "file:C:/dev/Java/
  javapro/jaas/client.jar" {
  permission javax.security.
    auth.AuthPermission
    "createLoginContext";
  permission javax.security.
    auth.AuthPermission "doAs";
  permission
    java.io.FilePermission
    "ProtectedFile.txt", "read";
};
```

and the relevant portion of the JAAS security policy file is shown here:

```
// note: client_action.jar
// contains
// AuthorizedAction.class
grant codebase "file:
  C:/dev/Java/javapro/jaas/
  client_action.jar", Principal
  javapro.jaas.PrincipalImpl
  "tmodi" {
  permission
    java.io.FilePermission
    "ProtectedFile.txt", "read";
};
```

The Java security policy file grants three permissions to the AuthenticatedClient class: the permission to create a login context, the permission to invoke the doAs() method on a Subject, and the permission to read a file called ProtectedFile.txt in the current directory.

The JAAS security file looks similar to a regular Java Security Policy file, except that it associates a Principal with a group of permissions, as well. Per the policy files shown, the principal "tmodi" is allowed to read a file called ProtectedFile.txt in the current directory. A JAAS policy file can be specified by using the java.security.auth.policy system property.

Because JAAS extends the standard Java security infrastructure, both policy files have to grant read access to the file called ProtectedFile.txt in the current directory.

Listing 6 A Big Committment

```
public boolean commit() throws LoginException {
  if (succeeded == false) {
    return false;
  } else {
    // add a Principal (authenticated identity)
    // to the Subject

    // assume the user we authenticated is the
    // SamplePrincipal
    userPrincipal = new PrincipalImpl(username);
    if (!subject.getPrincipals().contains(
      userPrincipal))
      subject.getPrincipals().add(userPrincipal);

    // in any case, clean out state
    username = null;
    for (int i = 0; i < password.length; i++)
    password[i] = ' ';
    password = null;

    commitSucceeded = true;
    return true;
  }
}
```

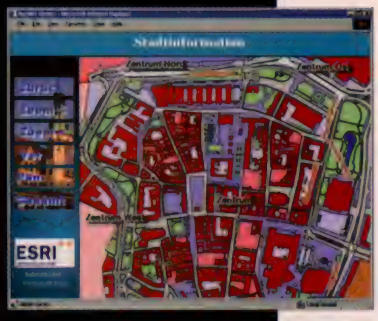
In our application, if all the login modules return a true boolean value, then the login context starts calling the commit() method on each login module. Here is our commit() method.

Listing 7 Abort Mission

```
public boolean abort() throws LoginException {
  if (succeeded == false) {
    return false;
  } else if (succeeded == true &&
    commitSucceeded == false) {
    // login succeeded but overall
    // authentication failed
    succeeded = false;
    username = null;
    if (password != null) {
      for (int i = 0; i < password.length; i++)
        password[i] = ' ';
      password = null;
    }
    userPrincipal = null;
  } else {
    // overall authentication succeeded and
    // commit succeeded, but someone else's
    // commit failed
    logout();
  }
  return true;
}
```

Here is the abort() method for our login method.

Delivering a New World of GIS to Your Desktop



ESRI's powerful Internet mapping system, ArcIMS™, provides a framework for distributing and disseminating GIS services on the Internet. ArcIMS makes it easy to create, design, and manage Internet sites that incorporate e-mapping and e-GIS capabilities.

ArcIMS Features

- Fast, dynamic mapping on very large data sets
- Ability to leverage your data investment by publishing maps on the Web
- Easy-to-use, Web browser-based interface
- Simple installation, implementation, and administration with wizards and templates
- "Site Starter" templates to get you up and running fast
- Unique integration of local and Internet data
- Highly scalable architecture
- Powerful clients including free Java™ viewer that supports streaming vector data
- High-quality cartographic rendering
- Advanced mapping tools
- Geocoding capabilities

ArcIMS™

The GIS for the Internet

www.esri.com/arcims

ESRI

1-888-289-5084
info@esri.com

Copyright © 2001 ESRI. All rights reserved. ESRI and the ESRI globe logo are trademarks of ESRI, registered in the United States and certain other countries; registration is pending in the European Community. ArcIMS and the ArcIMS logo are trademarks and @esri.com and www.esri.com are service marks of ESRI. Microsoft and the Windows logo are registered trademarks and the Microsoft Internet Explorer logo is a trademark of Microsoft Corporation. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

Listing 8 Log Out

```
public boolean logout() throws LoginException {

  subject.getPrincipals().remove(userPrincipal);
  succeeded = false;
  commitSucceeded = false;
  username = null;
  if (password != null) {
    for (int i = 0; i < password.length; i++)
      password[i] = ' ';
      password = null;
  }
  userPrincipal = null;
  if(debug)
    System.out.println("logged out.");
  return true;
}
```

The logout() method cleans out the private member variables.

In other words, the permissions in the JAAS policy file will always be a subset of the permissions in the Java security policy file. Furthermore, although the JAAS security policy is physically separate from the existing Java 2 security policy, the two policies, together, form one logical policy.

Use the login context to perform the logout It is always good practice to log out as soon as the user-specific actions have been performed, or at least before exiting the application. The client does so by calling the logout() method on the login context:

```
try
{
  lc.logout();
}
catch (LoginException e) {
  System.out.println(
    e.getMessage());
}
```

Calling the logout() method on the login context results in the logout() method on the login module being called. When a client uses the login context to perform a login or logout, the calls are ultimately forwarded to at least one login module.

Implementing the Login Module

A login module implements the authentication algorithm. Because we want to implement our own login module, we must decide on an authentication scheme. Without becoming too fancy, I vote for a simple username/password scheme with the username/password pair being stored in a persistent file. To make matters slightly more interesting, the password in the file is the message digest of the real password selected by the user. Therefore the real password is not stored in the system. As an aside, the message digest algorithm I will use is called MD5.

To make life easier, I have created a program called PwdUtil.java that allows you to create new password files, load existing ones, and add, remove, and change existing users and passwords. PwdUtil.java and PasswordBean.java, which shows a helper bean that the utility uses, are available online. These programs are straightforward, so I will not go into a discussion about them. Listing 4, however, shows an example of how to use the password utility to create a new password file. This is the same passwords file that I specified in the options section of the configuration block in the JAAS configuration file.

All login modules must implement the LoginModule interface

```
public interface LoginModule {
  boolean login();
  // 1st authentication phase
  boolean commit();
  // 2nd authentication phase
  boolean abort();
  boolean logout();
}
```

JAAS supports the notion of stacked login modules. We saw this in the form of multiple login modules per configuration block in the JAAS configuration file. To guarantee that either all login modules succeed or none succeed, the login context performs the authentication steps in two phases. In the first phase, or the "login" phase, the login context invokes the configured login modules and instructs each to attempt only the authentication process. If all the necessary login modules successfully pass this phase, the login context then enters the second phase and invokes the configured login modules again, instructing each to formally "commit" the authentication process. During this phase, each login module associates the relevant authenticated principals and credentials with the subject. If either the first phase or the second phase fails, the login context invokes the configured login modules and instructs each to abort the entire authentication attempt. Each login module then cleans up any relevant state it had associated with the authentication attempt.

Now let's take a look at the implementation of our login module. We start by looking at the initialize method, which, if you recall, is called during the creation of the login context by JAAS. The initialize method gets a reference to a Subject object and a callback handler. In our case, the Subject object will be empty (it does not contain any authenticated principals), and the callback handler will be a reference to the MyCallbackHandler object (because that is what we passed into the login context during its creation). An interesting part of this method is how it gains access to the optional fields specified in each line of the cofiguration block. Turn back to our configuration file for a moment and look at this line:

```
javapro.jaas.LoginModuleImpl
  required
  debug=true
  fileName=passwords;
```

Here the parts debug=true and filename=passwords are the optional fields. Now look at this code, which is used to access them:

```
// initialize any configured
// options
debug = "true".equalsIgnoreCase((
  String)options.get("debug"));
try
{
  passwords.load((
    String)options.get(
    "fileName"));
  if(debug)
```

The Hottest Magazine for the Hottest Developers in the Enterprise

If you're a professional developer programming in the hottest language in the fastest growing development market in the universe, then you need hands-on, code intensive, product-oriented information that comes only from *Java Pro* magazine. Harness Java's unique capabilities in YOUR enterprise with no-holds-barred editorial content.

Don't miss this special offer! Get 13 information-packed issues—a full year of Java Pro *—for only $19.95. That's a savings of more than 74% off the cover price. Call Today!*

Java™ Pro: get beyond the hype

- Practical techniques to amplify Java performance
- Independent, objective product reviews to help you find only the best products for your needs
- Complete source code examples using JavaBeans, servlets and more
- Tips and tricks for Java power programmers

***Java Pro:* is the professional developers independent source for building serious Java applications**

Order online today
www.javapro.com

Subscribe by phone
1-866-387-5776

FTP FAWCETTE TECHNICAL PUBLICATIONS
209 Hamilton Ave.
Palo Alto, CA 94301

One year is thirteen issues. Annual cover price is $77.35. Subscription begins upon receipt of payment. Please allow four-to-six weeks for delivery of first issue. International subscriptions must be prepaid in U.S. dollars plus postage. Canada/Mexico, add $18/year; other countries add $44/year for airmail. Canadian GST included. *Java Pro* is published by Fawcette Technical Publications, Inc. Java is a trademark of Sun Microsystems, Inc.

Listing 9 Canning Methods

```
REM Assume that the JDK is installed in
  C:\jdk1.3
REM Assume that the Java source files are
REM in the directory C:\dev\Java\javapro\jaas

REM Compile the Java source code
C:\jdk1.3\bin\javac -classpath C:\dev\Java\
  -d ..\..\ PasswordBean.java PwdUtil.java
  PrincipalImpl.java LoginModuleImpl.java
  AuthenticatedClient.java AuthorizedAction.java

cd ..\..

REM JAR up the client classes
C:\jdk1.3\bin\jar cvf client.jar
  javapro\jaas\AuthenticatedClient.class
  javapro\jaas\MyCallbackHandler.class

REM JAR up the action class
C:\jdk1.3\bin\jar cvf client_action.jar
  javapro\jaas\AuthorizedAction.class

REM Jar up the login module classes
C:\jdk1.3\bin\jar cvf loginmodule.jar
  javapro\jaas\LoginModuleImpl.class
  javapro\jaas\PrincipalImpl.class
  javapro\jaas\PasswordBean.class

move *.jar javapro\jaas\
cd javapro\jaas
```

Once you have copied the jaas.jar file into your extension directory, you can compile the programs and create these JARs.

```
    System.out.println(
      "Passwords file: " +
      (String)options.get(
      "fileName"));
}
catch(Exception e)
{
  if(debug)
    System.out.println(
      e.getMessage());
}
```

As you can see, these items are available as name-value pairs in the options parameter passed to the initialize method. When the debug option is set to "true", the login module prints informational messages on standard out. The passwords option is mandatory (so I guess calling it an option is a misnomer). This is where the login module gets all the information about valid usernames and their passwords (actually a message digest of each password). This file can be created with the password management utility already discussed.

Again, JAAS authentication is a two-step process. Remember, the whole process of authentication is kicked off when the client calls the login method on the login context. During the first step, the login context calls the login method on the login module. Let's look at our implementation of this method. First, the method uses the callback object to get the username and password from the client requesting the login (see Listing 5). Read the inline comments to understand the logic.

Once the username and password are retrieved, the authentication is a simple matter of verifying that such user does exist in the passwords file and that the message digest of the password matches exactly the message digest of that user's password in the passwords file:

```
// verify the username/password
try
{
  String correctPasswordHash =
    passwords.getPassword(
    username);
  if(correctPasswordHash == null)
    throw new LoginException(
      "Error: No such user
      exists.");
  String hash =
    passwords.getPasswordDigest(
    new String(password));
  if(!hash.equals(
    correctPasswordHash))
    throw new LoginException(
      "Error: Incorrect
      password.");
}
```

If this authentication succeeds, then the method returns a true boolean value, which is when the second step begins. If all the login modules (in our case just one) return a true boolean value in their respective first step, then the login context starts calling the commit() method on each login module. If even one of the login modules returns a false value, then the login context calls the abort() method on the login modules. Note the striking resemblence to the two-phase commit protocol used by transaction coordinators for commiting transactions.

Let's take a look at the commit() method (see Listing 6). The succeeded flag is set to true by the login method if it is successful. Theoretically the commit() method should not be called if the succeeded flag is false, but in any case it does not hurt to check. The only task now is to create a new principal that wraps the user (username in our case) and add it to the list of authenticated principals to the Subject object reference in the module. The principal class, PrincipalImpl.java (available online), is a simple class that implements the Principal interface. Once that is done, we clean up the private member variables that store the username and password because they are no longer required. A golden rule to remember while creating and implementing security-sensitive applications is to get rid of any information, no matter how insignificant it may seem, as soon as it is not required.

Now let's take a look at the abort() method (see Listing 7). The inline comments are self-explanatory. We abort because we encounter one of three situations:

- Our login() method returns a false. In that case we just return.
- Our login() method returns a true, but the commit method fails. In that case we need to clean out the username and password fields.
- Both our login() and commit() methods succeed, but another login module along the chain fails. That means that as far as our module goes, we're logged in, so we must call our logout() method.

So that brings us to the last method of our login module, logout() (see Listing 8). Basically it cleans out the private member variables. Because our module does not really "login" anywhere, it does not have to "logout," either. LoginModuleImpl.java (available online) shows the complete implementation.

Bringing it all Together

Now let's check out our program. Make sure that you have copied the JAR file jaas.jar (available as part of the JAAS installation) in your extensions directory. If you have not, then you must include it in your classpath. Now compile the programs and create the JARs as shown in Listing 9. Next, run the authenticated client as shown in Listing 10 online.

When you get prompted for a username and password, enter "tmodi" and "a" respectively (without the quotes). Everything should work. Run the program again and enter "mala" and "123" as the username and password, respectively, and you should get an access exception. This is because, per the JAAS policy file, only the user "tmodi" is authorized to "read" the file ProtextedFile.txt.

Single Sign-On

Single sign-on refers to the ability of an application to realize the credentials of a user already logged on to the native platform on which the application is running and use them. For example, when I'm logged on to my NT machine and I start Microsoft Outlook, it does not ask me to log in. Instead it recognizes that I'm already logged in to NT and uses those credentials to access my mailbox. This is single sign-on in action. Sun Microsystems has implemented login modules for both the Solaris and NT operating systems that let you add the single sign-on ability to your applications, as well. For example, making your authenticated client a single sign-on application is fairly simple. All you need to do is change the jaas.config file so that it uses the NT JAAS module from Sun Microsystems (or any other vendor) and the jaas.policy file to reflect the appropriate principal class and principal name.

Even though I have only scratched the surface of the endless possibilities with JAAS, it is obvious that JAAS provides a much needed extension to Java's existing security infrastructure; namely the ability to authenticate users and authorize code execution based on the authenticated user identity. It's easy to see how JAAS can be used to implement a role-based security model similar to the one available in Enterprise JavaBeans today. JAAS is slated to be part of version 1.3 of the J2EE platform, and it is likely that future versions of J2EE-compliant application servers will use JAAS under the covers to implement role-based security in a much more standard and portable way. JP

GO ONLINE
www.javapro.com

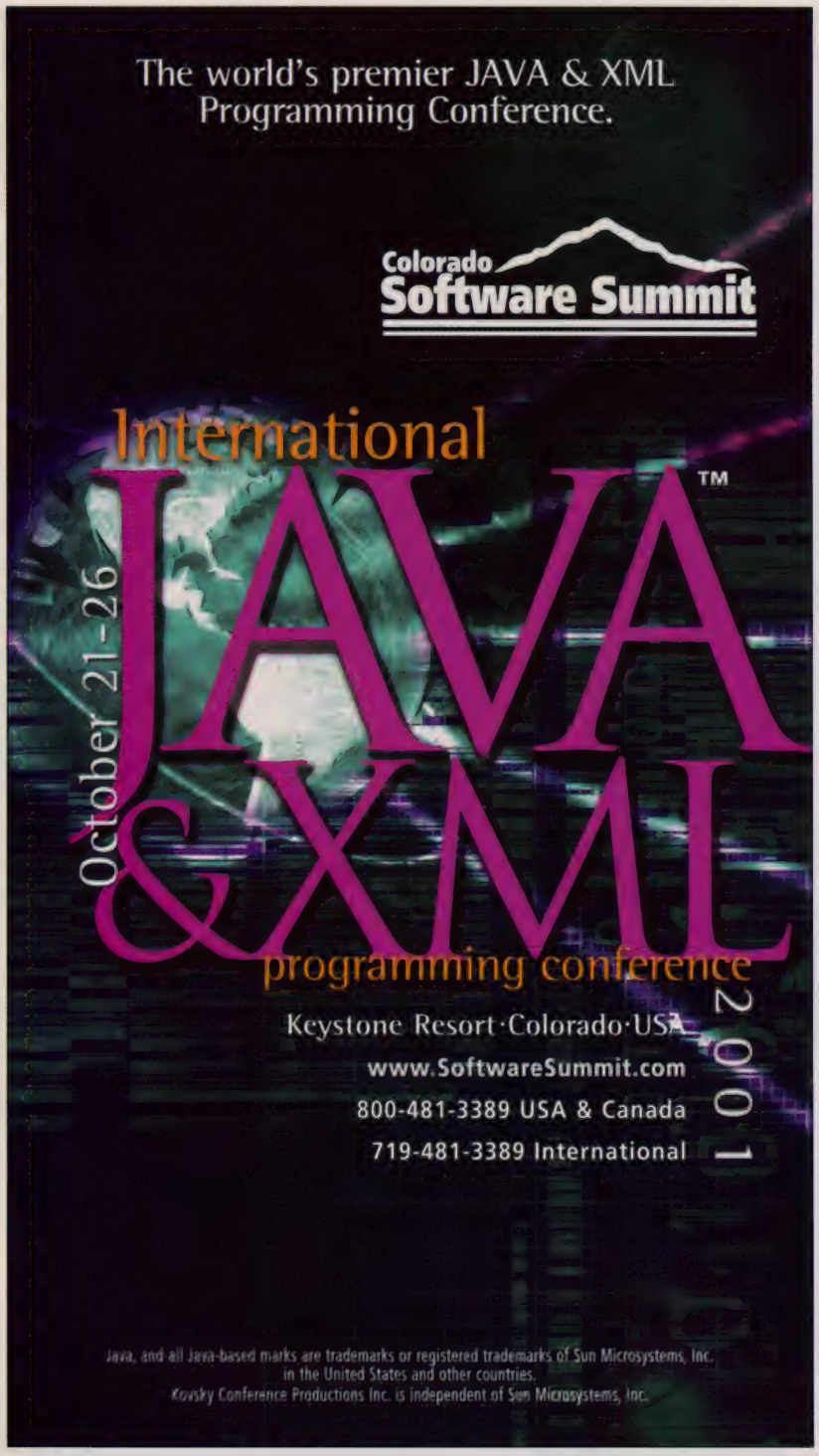
Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109TM Download the code for this article.

JP0109TM_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.


The world's premier JAVA & XML Programming Conference.
Colorado Software Summit
International
JAVA & XML
programming conference
October 21-26
2001
Keystone Resort·Colorado·USA
www.SoftwareSummit.com
800-481-3389 USA & Canada
719-481-3389 International
Java, and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.
Kovsky Conference Productions Inc. is independent of Sun Microsystems, Inc.

A Content Management and Delivery Solution Built on a

J2EE Application Server is Not Something You See Everyday.

So, how do you know when you've got the real deal?
Only if you can answer "YES" to ALL of these questions...

- √ Is your CMS application written in Java™?
- √ Does your application require a J2EE™ application server to run?
- √ Does your application provide a native Java interface?
- √ Is your entire application logic built and packaged as a set of servlets running natively on a J2EE application server?
- √ Does your application use JDBC and the connection pooling capability of the J2EE application server?
- √ Is your application built to use the clustering and failover capability of the J2EE application server?
- √ Does your application provide JSP support in process?
- √ Can your application leverage J2EE-based XML services?
- √ Does your application support the use of EJBs for custom business logic?
- √ Can your application be moved from one vendor's J2EE application server to another as business needs change?

A pure J2EE application is a rare and wonderful thing and an unqualified YES to each of these questions is the real acid test. So, ask your content management vendor and then ask Open Market — the industry's **only** content management and delivery vendor that can say **"YES!"** with complete confidence.

Open Market: The Real Deal.

Open Market: 100% Pure Java™ Content Management and Delivery for the industry's leading application servers: BEA WebLogic Server, IBM WebSphere Application Server and iPlanet Application Server.

Download an evaluation copy of Open Market's Content Server Enterprise Edition at:
www.openmarket.com/jumpstart

OPEN MARKET
1.800.OPEN-MKT

bea

IBM Business Partner

iPlanet e-commerce solutions A Sun|Netscape Alliance ISV PARTNER

Open Market is a registered trademark and Content Server is a trademark of Open Market, Inc. in the United States and other countries. Java, iPlanet, and all Sun, Java and iPlanet based trademarks are trademarks or registered trademarks of Sun Microsystems, Inc, in the United States and other countries. Bea and WebLogic are registered trademarks and BEA WebLogic Server is a trademark of BEA Systems, Inc. IBM and WebSphere are trademarks of International Business Machines Corporation in the United States and other countries.

How to Enhance Code Reusability

by Adam Kolawa

Currently, developers looking to exchange code are turning to components that frame functionality. You can easily plug third-party components into your application to quickly add basic functionality without reinventing the wheel. Likewise, you can make your own creations available to others who want to implement the technology that you have already perfected.

The problem with component exchange is that in order for components to be reused, they must be robust. You simply cannot reuse a component that runs flawlessly in its native environment, but crashes when it is used in new systems.

You can increase the chances of your components functioning flawlessly in unknown, unanticipated environments by ensuring that they can handle a wide variety of possible inputs—even unexpected inputs. One effective way to assess what your code can handle is to perform white-box (construction) testing on your classes. To perform white-box testing, examine the class's internal structure, determine what inputs will fully exercise the class, then run the test cases you've created to determine if there are any possible class usages that will make the class crash.

Use white-box testing to expose the limits of your code, then evaluate the limitations found before you take your next step. If you find that your code cannot handle certain inputs that it should be able to handle, modify the code so that it works correctly with the given inputs. If you find inputs that the code simply cannot and should not handle, or if you find expected exceptions, thoroughly document this information so that the requirements will be as clear as possible. This way, anyone using your code can clearly see the code's requirements and you can help prevent software reuse problems such as the infamous Ariane crash.

Adam Kolawa, Ph.D., is Chairman and CEO of ParaSoft. You can reach him at ak@parasoft.com

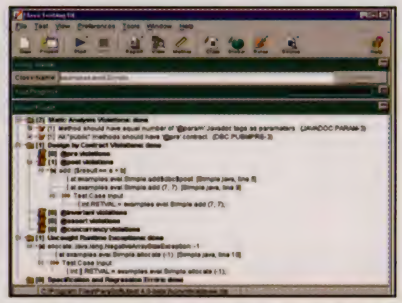
Now Automates Black-Box Testing!

ParaSoft® jtest®

Click once for automatic Java™ unit testing!

"It is fundamental that every piece of Java code is tested at the class level as soon as it's been written, but writing test cases is tedious and time consuming. Jtest lets me do this automatically and gives me the confidence that my code has been thoroughly tested. I love it!"
—Mansour Raad, DiscoverCase

The development community endorses unit testing, coding standards, and Design by Contract™ because they increase software reliability. Until now, these practices were too costly and labor-intensive to implement.



Jtest is user-friendly; you can start finding bugs automatically at the click of a button.

Jtest® eliminates these obstacles, giving even the most time-stricken developer the power to automate these critical procedures, including—for the first time in software development history—fully automated black-box testing. Jtest is an automatic unit testing tool for Java. With the click of a button, Jtest automatically performs the critical testing procedures discussed below, slashing debugging time and taking testing to the next level.

Black-box (functionality) testing ensures that methods in your class function properly. Jtest fully automates this by reading the specification information built into the class with the Design by Contact (DbC) language, then automatically creating and executing test cases that verify the functionality described in the specification. You can also add test cases by adding method inputs for simple test cases, or creating Test Classes for complex ones.

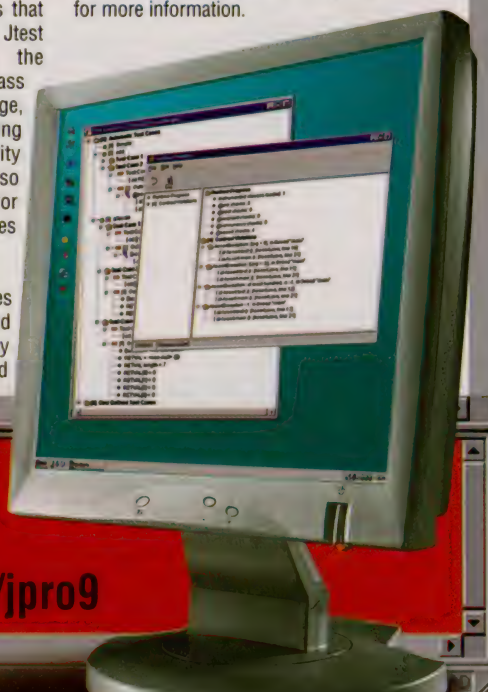
White-box (construction) testing produces robust classes that can withstand punishment. Jtest automates this by generating and executing test inputs based on a class's structure, ensuring unexpected inputs do not cause crashes. When an input generates an uncaught runtime exception, Jtest reports the arguments, stack trace, and calling sequence, pinpointing an error's exact location.

Regression testing verifies that code performs according to specification after changes are made. Jtest performs automatic regression testing at the class level, monitoring your existing code's integrity earlier than ever thought possible. To do this, simply restore a test and click a button or integrate batch-mode Jtest into your nightly builds to ensure that errors are found quickly.

Jtest prevents errors by automatically checking whether your code adheres to over 240 industry-respected coding standards. You can extend and customize Jtest's coding standards with the RuleWizard™ feature, which lets you graphically create custom standards.

Jcontract™ is a new tool that checks DbC contracts at runtime. It is complementary to Jtest and can be used to verify that code works correctly at the system level.

To use Jtest and Jcontract for your Java projects, download a demo today at www.parasoft.com/jpro9 or call (888) 305-0041 for more information.



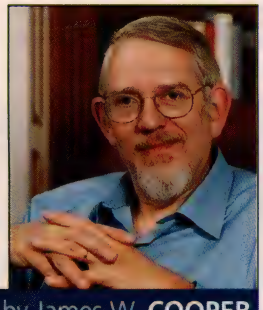
ParaSoft®

Download your **FREE DEMO** at:
www.parasoft.com/jpro9

© 2001 ParaSoft Corporation, Monrovia, CA, USA. ParaSoft and Jtest are registered trademarks of ParaSoft Corporation. RuleWizard and Jcontract are trademarks of ParaSoft Corporation. All other brands, services and companies are trademarks, registered trademarks or servicemarks of their respective holders in the U.S. and/or other countries.

A Walk in the Clouds

Use Cloudscape and database design patterns to build a scalable cross-platform solution in just minutes



by James W. **COOPER**

James W. Cooper is a computer science researcher and the author of 14 books. His latest, *Java Design Patterns: A Tutorial*, was published by Addison-Wesley in January 2000.

Java is portable: Write once, run mostly anywhere, right? But a persistent problem has been one of the portability of the underlying system tools you need to build complete applications on either the client or the server. One of the biggest headaches is handling databases, which despite their ubiquity have a number of annoying platform or price restrictions. At the low end is Microsoft Access, which although perfectly suitable for small applications, is plagued by being Windows-only and emphatically not multi-user. In the mid range, you have Microsoft SQL Server, which is certainly multi-user, but is still Windows-only and not inexpensive. At the high end, industrial-strength systems like DB2 and Oracle cover most platforms, but are kind of expensive and overkill for simpler applications.

Here is where the remarkable Cloudscape package comes into play. Cloudscape is an all-Java database system originally available from Informix (recently acquired by IBM) that runs anywhere you can run a JVM. Not only does this mean it can run on any Unix as well as any Windows system, it means it can run on Mac as well. This opens up the sophisticated user application market that needs a database to keep track of information. This information could be customer order info, a complete mailing list system, or a system that keeps track of domain-specific information like athletic times or genealogy information. Any system you built would run anywhere people want it to. It's easy to provide it to other potential users. You would not be limited to PCs, Macs, or Linux. Finally you can write a truly cross-platform user application!

Of course these generalizations also apply to server systems. You can do JavaServer Pages (JSP)-based lookups of any kind of information on your server, and be sure that the system will port to other platforms without any significant reprogramming. This can be ideal for locally developed systems that other related groups want to implement without the interference of religious issues. Finally, if you have written your server applications in Java, Cloudscape provides a clear migration path to one of the industrial-strength database systems when your system demands increase.

Trying Out Cloudscape

The complete, working version of the Cloudscape developer kit is available at www.cloudscape.com/

Listing 1 Generate a Statement

```
public void createTable() throws SQLException {
  if (exists() )  //delete previous table
    delete();
  //generate SQL string to make the table
  String sql = "CREATE TABLE " +
    tableName +" (" ;
  sql += getFieldString();    //get column names
  if (primaryKey.length () > 0) {
    sql += ", PRIMARY KEY(" + primaryKey +")";
  }
  sql +=")";
  SqlQuery qry = new SqlQuery(sql, db);

  qry.executeUpdate ();
  tableCreated = true;
}
```

In our previous discussion of the Façade pattern, we talked about accessing databases and getting results. Then we created a new class (DBTable) that allowed us to create new tables in addition to querying existing data in a database. The DBTable class generates the SQL CREATE TABLE statement. The important method is shown here.

Listing 2 Table Store

```
private void makeStores() throws SQLException {
  System.out.println("creating: Stores");
  stores = new DbTable("Stores", db);
  stores.addField ("Storekey");
  stores.addField("StoreName", 20);
  stores.setPrimaryKey ("Storekey");
  stores.createTable ();
  stores.openTable ();

  int i = 1;
  addStore(i++, "Village Market");
  addStore(i++, "Stop and Shop");
  addStore(i++, "Waldbaums");
}
```

We want to create a database containing three tables: Stores, Foods, and FoodPrices. Here we see the method for creating the Stores table.

Listing 3 Table of Groceries

```
private void makeGrocs() throws SQLException {
  System.out.println("creating: Food");
  grocs = new DbTable("Food", db);

  grocs.addField ("Foodkey");
  grocs.addField("Foodname", 20);
  grocs.setPrimaryKey ("Foodkey");
  grocs.createTable ();
  grocs.openTable ();

  int i = 1;
  addGroc(i++, "Apples");
  addGroc(i++, "Oranges");
  addGroc(i++, "Milk");
  addGroc(i++, "Butter");
  addGroc(i++, "Green beans");
  addGroc(i++, "Cola");
  addGroc(i++, "Hamburger");
}
```

Our second table has some grocery names added to it.

Listing 4 Database Creation

```
//creates the tables in the database
private void createDB() throws SQLException {
  Cursor wait = new Cursor(Cursor.WAIT_CURSOR);
  setCursor(wait);
  db.create("GroceryDB"); //Create the databse
  makeGrocs(); //groceries table

  makeStores(); //stores table
  makePrices(); //food price table
  Cursor def = new Cursor(Cursor.DEFAULT_CURSOR);
  setCursor(def);
  db.close (); //close
}
```

The database creation step in our grocery program uses methods we have already shown.

products/fdkinfo.jsp. It is free for personal use and for development. When you are ready to deploy a Cloudscape-based system, the company will provide a package price that depends on the number of users and systems on which you will be deploying.

I got a copy of the free Cloudscape 3.6 CD-ROM and installed the program on my Windows 2000 laptop in just a few minutes. The only confusion that arose was a requirement that the directory path not contain any spaces. You can work around this manually, however. Once Cloudscape is installed, you find that it comes with excellent documentation and a very well written and designed tutorial, as well as example code you can run on the spot to see how Cloudscape works. I read the whole tutorial and soon realized that I was going to be able to write code to build and use Cloudscape very quickly indeed because of the design patterns we've discussed in previous columns.

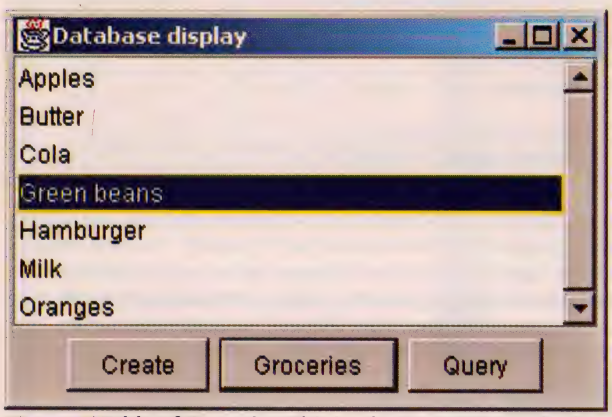


Figure 1 | Table of Groceries This is what our program looks like after the grocery table is created.

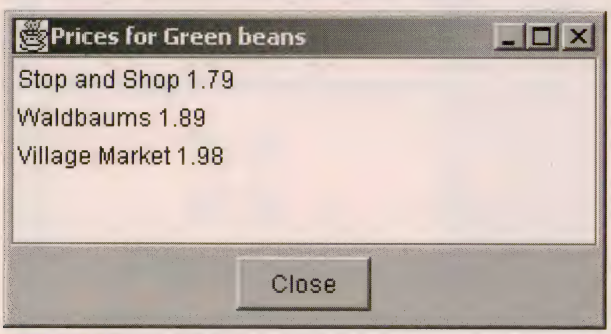


Figure 2 | Bargain Shopping Here are the results of our price query.

You can access Cloudscape directly through its SQL interface and that is what I did in writing the code for this column. In fact, I spent a lot more time making a user interface to display my examples than I did programming to Cloudscape because everything worked right out of the box the first time. In addition to the SQL interface, Cloudscape comes with Cloudview, which allows you to view and modify any database from a convenient set of GUI windows.

The Façade Design Pattern

I originally discussed the Façade pattern in December of 1998 and in more detail in my book on Java design patterns. Then I extended its use in an article in October of 2000. Don't worry if you can't lay your hands on these issues right away, all of the example code is available for download with this article at www.javapro.com.

FSKey	StoreKey	FoodKey	Price
1	1	1	$0.27
2	2	1	$0.29
3	3	1	$0.33
4	1	2	$0.36
5	2	2	$0.29
6	3	2	$0.47
7	1	3	$1.98
8	2	3	$2.45
9	3	3	$2.29

Table 1 | Price of Food Here we see a portion of the FoodPrice table.

The Façade design pattern wraps some fairly tricky-to-use classes in some simpler ones, perhaps with a little bit less flexibility. The idea is really just another instance of the old 80/20 rule: Most users use only about 20% of the function, so why not make it easier for them. The more elaborate features are used seldom, but are available by going to the lower level interface when needed.

In the case of the java.sql.* package, the code for accessing databases and getting results is just a little too hard to use. It requires

Multi Platform Installations for the Real World

Real World Experience. Real World Partners. A Real Industry Leader.

Let's face it, software development has changed. Long past are the days of single-platform development — the world of the different and diverse is your reality. Where once there was a single operating system, now there are six. Where once there was a clear-cut release cycle, now there's complexity, maybe even a little chaos. Sure, all this diversity is great — if only it didn't cause so much pain.

We understand your world, and have created something to make it a lot easier. InstallShield MultiPlatform simplifies. It manages the diversity. It lets you create a first-rate installation without dealing with the hassles of platform-this and platform-that. After all, you want to ship software that thrills your customers — not get mired in technical details.

The magic behind InstallShield MultiPlatform is no secret — it's our partnerships. We realized early on that to create the world's best installation technology, we'd have to work closely with platform vendors like IBM, Sun Microsystems, and Microsoft. And that's exactly what we've done. It's the result of the experts working together that makes this product so effective.

When you think about installations, on whatever the platform, go with a real industry leader. As the pioneer of user-friendly, wizard-based installations, InstallShield has been adopted by all of the top 100 ISVs. We're the eigth most used software in the world with millions of users. And with over a decade of experience installing software, you can be assured that, whatever you're facing, we've got it covered.

So if you're pressured by the world of PCs, RS/6000s, Sun Servers, AS/400s, or that new mystery box that arrived last week, relax... InstallShield MultiPlatform was designed for just such a world.

key features

InstallShield MultiPlatform offers hundreds of features to improve your productivity as a developer and to guarantee the success of your product installations. By working closely with developers like you, we've ensured that each feature in InstallShield MultiPlatform addresses real-world issues. Here are just some of the features you will find:

- High Productivity IDE
- Component Based Development
- Full Automated Build Support
- Integrated Component Gallery
- XML/XSL Project Features
- Multiple Build Configurations
- Graphical UI and Console Support
- Extensive Language Support
- Conditional Wizard Sequences
- Product Branding Options
- Cross-Platform Product Registry
- Advanced Java Support
- Failure Detection and Rollback
- Advanced Event Logging
- Common Information Model (CIM)
- Over 140 Cross-Platform Functions
- Native Installation Compatibility
- Support for Six Operating Systems

GET YOUR FULL-FEATURED EVALUATION AT WWW.INSTALLSHIELD.COM

2001 JAVAPro READER'S CHOICE WINNER

JDJ EDITOR'S CHOICE AWARD

JAVA REPORT READERS' CHOICE

JDJ READERS' CHOICE AWARD

© 2001 InstallShield Software Corporation. All rights reserved. InstallShield is a registered trademark and service mark of InstallShield Software Corporation. All other trademarks are the property of their respective owners.

InstallShield® SOFTWARE CORPORATION

Listing 5 Load the Table

```
private void showGrocs() {
  try {
    db.Open ("GroceryDB");
    String sql="Select FoodName from Food
      order by FoodName";
    SqlQuery qry = new SqlQuery(sql, db);
    Results rs = qry.Execute ();
    while (rs.hasMoreElements ()) {
      jlist.add (rs.getColumnValue (1));
    }
  } catch (SQLException e) {
    System.out.println(e.getMessage ());
    e.printStackTrace ();
  }
}
```

This query loads the table when you click on the Groceries button.

Listing 6 Issue a Query

```
String queryText ="SELECT FoodName, StoreName,
  Price "+ "FROM (Food INNER JOIN FoodPrice ON "
  + "Food.FoodKey = FoodPrice.FoodKey) " +
  "INNER JOIN Stores ON " + "FoodPrice.StoreKey =
  Stores.StoreKey "+ "WHERE (((Food.FoodName)=
  \'" + produce+"\')) " + "ORDER BY
  FoodPrice.Price";
try {
  SqlQuery qry = new SqlQuery(queryText, db);
  Results rs = qry.Execute ();
  while (rs.hasMoreElements () ) {
    Double price = new Double(
      rs.getColumnValue (3));
    DecimalFormat form = new DecimalFormat();
    form.applyPattern ("##.00");
    jlist.add (rs.getColumnValue (2)+" "+
      form.format(price.doubleValue ()));
  }
}
catch(SQLException e) {
  System.out.println(e.getMessage());
}
```

Here we see the query that is issued when we click on the Query button.

understanding seven different classes, when most of the time two or three would seem to be all you need. So, I wrapped this complexity into three main classes:

- Database—handles connection to the database itself
- SqlQuery—packages each query and returns a Results class
- Results—returns the rows from a query

Then in the October 2000 column, I expanded on this in the case where we wanted to create new tables in addition to querying existing data. We created a DBTable class that allows you to create tables and add columns and indexes to them, and then add data to the tables. In essence, this class generates the SQL CREATE TABLE statement. The important method is shown in Listing 1. The getFieldString() method generates a list of the columns and types to be generated from the columns you define.

Suppose we want to create a database containing three tables: Stores, Foods, and FoodPrices. The method for creating the Stores table is shown in Listing 2 using the DBTable class. The addStore() method creates each row:

```
private void addStore(
  int key, String name) throws
  SQLException {
  System.out.println(
    "adding:" + name);
  stores.setPreparedTerm (
    key, 1);
  stores.setPreparedTerm (
    name, 2);
  stores.addRow();
}
```

Now, back to Cloudscape. The good news is that all of the code from these previous two columns worked without any significant change. However, Cloudscape's SQL parser is not happy with semicolons at the end of SQL statements. Because they are optional in other systems, I just deleted them from the DBTable class.

I added some grocery item names to a second table (see Listing 3) and made a price table in the same way using the

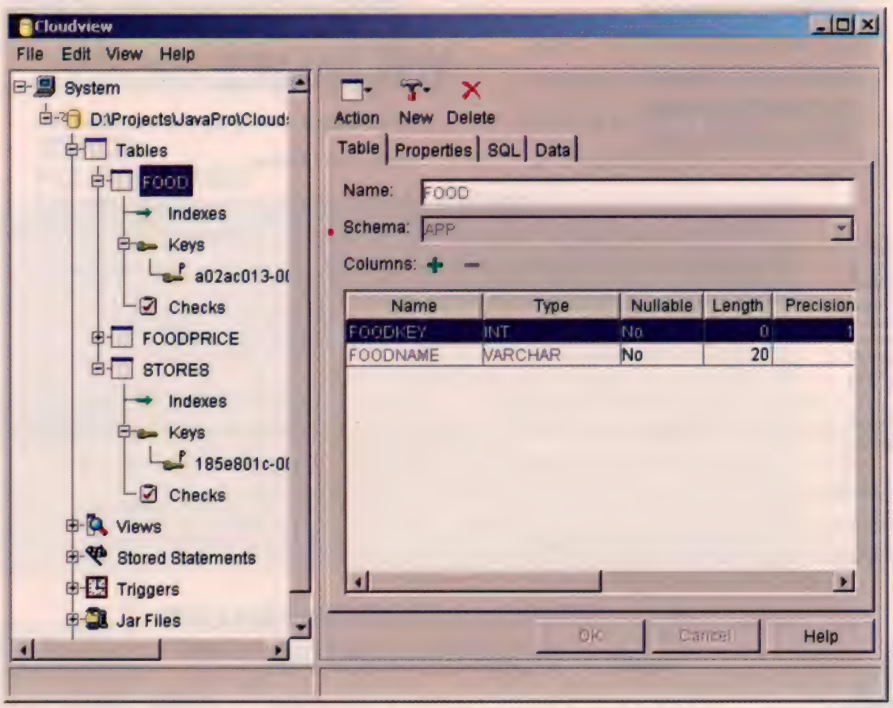


Figure 3 | View From a Cloud This is what the Food database table structure looks like in Cloudview.

THE ENTERPRISE JOB SCHEDULER

FLUX, the Enterprise Job Scheduler, is a Java software solution for job scheduling needs. Job scheduling is concerned with executing the correct task, on the correct day, and at the correct time.

One-time or recurring job scheduling

A single job can be executed at a specified time or repeatedly over longer intervals. Recurring jobs run indefinitely, until an ending date, or for a fixed number of repetitions.

Persistence

If desired, jobs can be stored in a relational database. Jobs will not need to be rescheduled when Flux is restarted.

J2EE and Java integration

Flux naturally integrates with J2EE and Java applications. Flux can be used as a software component in any J2EE or Java application to meet job scheduling requirements.

Flux now features job queuing, greater scalability, holiday and business calendars, job interruptions, pause & resume, an audit trail, and reporting.

To learn more about Flux, download a 30 day evaluation.

Sims Computing, Inc.
www.simscomputing.com

flux

simscomputing

+1 (406) 656-7398 info@simscomputing.com sales@simscomputing.com

Listing 7 Connecting Strings

```
//Connects to Cloudscape databases
public class CloudDatabase extends Database {
  public CloudDatabase()
    throws ClassNotFoundException {
    super("COM.cloudscape.core.JDBCDriver");
  }
  //-----------------------------
  public void create(String databaseName)
    throws SQLException {
    super.Open ("jdbc:cloudscape:"+
      databaseName+ ";create=true", null);
  }
  //-----------------------------
  public void Open(String databaseName)
    throws SQLException {
    super.Open ("jdbc:cloudscape:"+
      databaseName+ ";", null);
  }
  //-----------------------------
  public void close() {
    boolean gotSQLExc = false;
    try {
      DriverManager.getConnection(
        "jdbc:cloudscape:;shutdown=true");
    } catch (SQLException e) {
      gotSQLExc = true;
    }
    catch(Exception e) {
      System.out.println("Abnormal shutdown:"
        +e.getMessage());
    }
  }
}
```

To create a database using a special connect string that Cloudscape supports, you need to derive the CloudDatabase class from your generic Database class, as shown here.

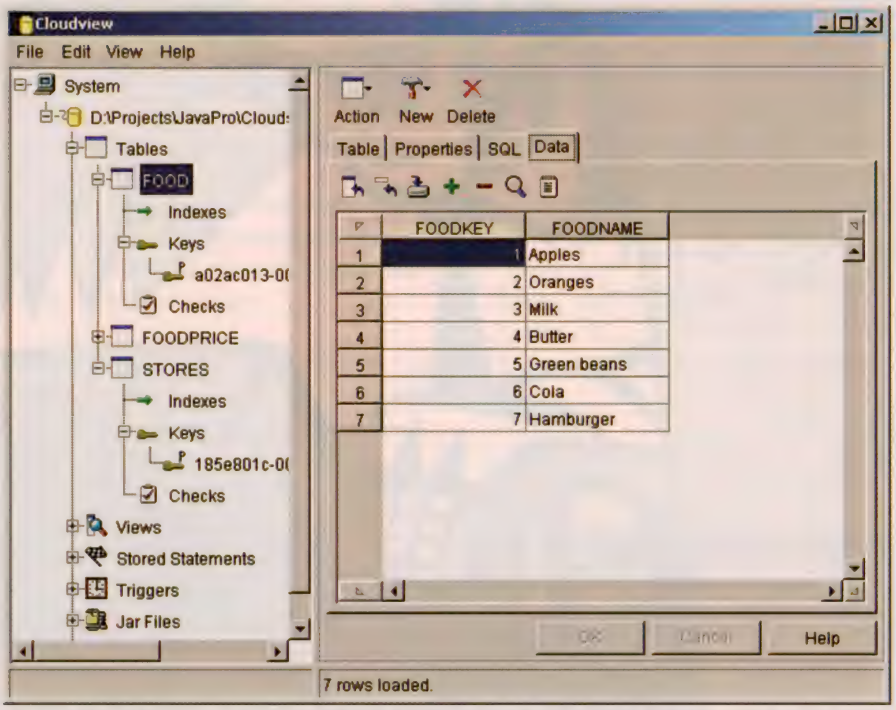


Figure 4 | Food Contents Shown in Cloudview, here are the contents of the Food database table.

StoreKey, the FoodKey, and a price value as shown in Table 1.

Our program creates the database each time and allows you to query the list of groceries and select one to get the price. Using the methods we have already shown, the database creation step becomes the code in Listing 4. Figure 1 shows the program after the grocery table is created. The query to load the table when you click on the Groceries button is shown in Listing 5. When we click on the Query button, we show the query window and issue the query in Listing 6. The results are shown in Figure 2. To see what we've created, I ran Cloudview to display the database tables (Figure 3) and the contents of one of the tables (Figure 4).

What Did I have to Change?

I really changed only two things to get Cloudscape working with my existing Façade code. First, I removed the semicolons from the end of all the SQL generated by the classes. Second, I derived the CloudDatabase class from my generic Database class to allow for the ability to create a database using a special connect string that Cloudscape supports (see Listing 7).

Other than that, Cloudscape and my classes worked together right from the beginning. I built a scalable cross-platform database solution in just a few minutes. I think this is pretty impressive. Cloudscape will handle multiple connections and it provides some example JSP pages for using it in a client/server environment. Because it is a native Java system, you can include stored Java object instances and stored Java procedures. I liked that a lot. Finally, should I need to scale to some humongous multiserver solution with terabytes of data, I can just plug in DB2 instead of Cloudscape without changing anything except the version of the Database class I use. I'm very impressed.

For completeness, I must point out that Cloudscape is not the only all-Java database product you can buy. Pointbase is also such a product, and from a cursory examination, it also appears to be well done. I'll report on it in a future column.

Editor's Note: IBM Corporation acquired all of the database assets of Informix Corporation in July 2001. As a result of the acquisition, IBM plans to integrate the Cloudscape technology into the DB2 family. Please refer to the following URLs for evolving details: www.cloudscape.com and www-4.ibm.com/software/data/db2/everyplace. JP

GO ONLINE
www.javapro.com

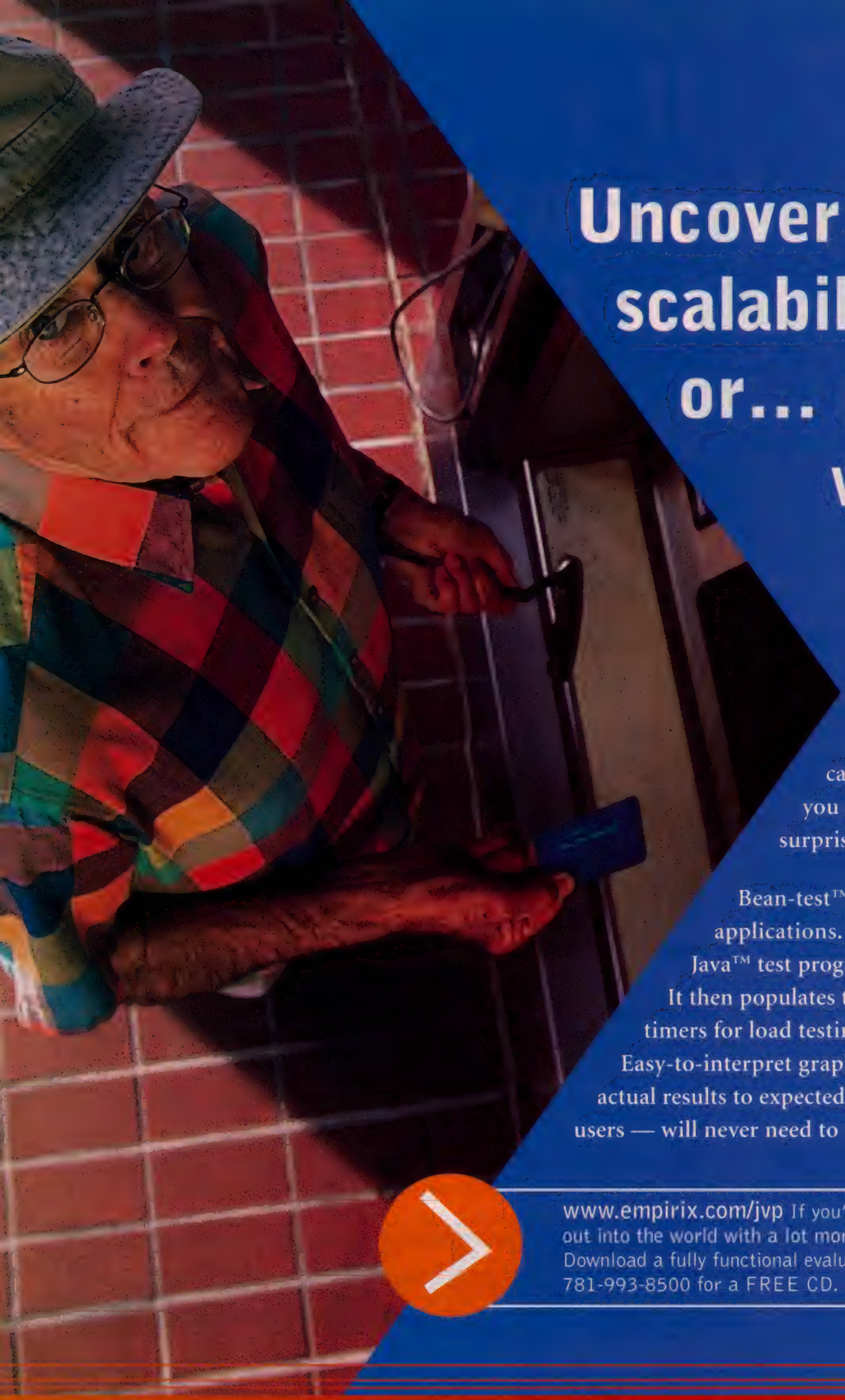
Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109JT Download the code for this article.

JP0109JT_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.



Uncover EJB scalability issues or... guess who will

Test the scalability and quality of your EJB™ middleware applications — before you go live — or you may be in for some unpleasant surprises down the road.

Bean-test™ lets you quickly test your EJB applications. It automatically generates a Java™ test program that tests your EJB methods. It then populates the test with data and establishes timers for load testing under real-world conditions. Easy-to-interpret graphs and logs let you compare actual results to expected results so you — and your end-users — will never need to worry.

www.empirix.com/jvp If you're ready to send your applications out into the world with a lot more confidence, try Bean-test today. Download a fully functional evaluation version of Bean-test or call 781-993-8500 for a FREE CD.

Bean-test makes sure your EJB applications handle the load.

Empirix is the leading source for test and monitoring solutions that measure end-to-end Quality of Experience (QoE) for communications infrastructure, Web and voice applications, contact centers and more. Empirix, established by combining Hammer Technologies and RSW Software, offers scalable, powerful testing products and services that reduce costs, shorten deployment time, enhance performance and strengthen revenue streams across today's rapidly converging e-communications channels.

www.empirix.com

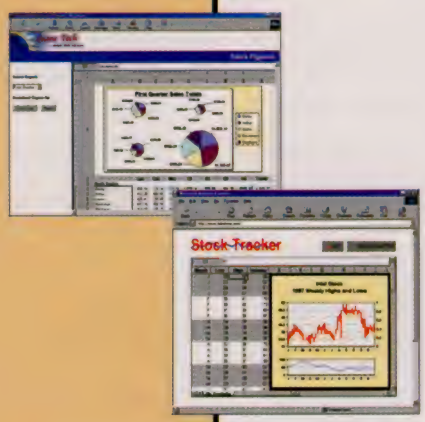
©2001 Empirix, Inc. All rights reserved. Bean-test, RSW Software, and Empirix are trademarks and Hammer Technologies is a registered trademark of Empirix, Inc. Java and EJB are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. All other trademarks are the property of their respective owners.

empirix

STRESSING PERFORMANCE

If your users want reports in Excel, you need Formula One for Java.

Deploy Actuate's Java-based reporting functionality from Actuate e.Reporting Server or any J2EE application server.



Excel-compatible Java spreadsheet engine for building reports & charts into Web apps

Actuate Formula One for Java automates and Web-enables the most common spreadsheet-based reporting and charting tasks in financial and calculation-intensive industries:

- Embed powerful reporting and charting functionality in JSPs, servlets, EJBs, and server-based Java applications.
- Dynamically generate richly formatted Excel reports over the Web with formulas, charts, formatting, outlining, and other interactive features — a stark contrast to the CSV or tab delimited files generated by most applications for Excel support.
- Provide a robust spreadsheet interface for presenting and manipulating data in applets deployed to browsers or Java desktop applications — with no reliance on Excel or Windows.

FORMULA ONE FOR JAVA

- Perform scalable calculations on servers by leveraging spreadsheet functions and formulas. No need to write your own algorithms or calculations.
- Deliver interactive Web charts with clickable areas for "drilling down" into data.
- Utilize existing Excel spreadsheets as templates for reports and charts instead of hard-coding reports in Java.
- Access XML, databases, text files, and other server-side data sources.
- Read and edit existing Excel spreadsheets in 100% Pure Java environments.

For DHTML, HTML, PDF, or WYSIWYG reports, try Actuate e.Report Designer Java Edition

Actuate e.Report Designer Java Edition enables developers to extract, format, and output data efficiently.

- Access data from Java objects inside applications, databases, EJBs, and text files. Format the data in many different ways with easy to use wizards and WYSIWYG editor.
- Present information using crosstabs, parameters, multiple sections, and grouping levels, charts and graphs, conditional formatting, and more.
- Deliver reports as DHTML, HTML, PDF, CSV, and a WYSIWYG hard-copy printout.
- 100% Pure Java.

Download Free 30-Day Trials

Try **Actuate Formula One for Java** or **Actuate e.Report Designer Java Edition** in your next reporting application **FREE** for 30 days. Visit **www.actuate.com/info/jpad.asp** today!

ACTUATE

800-884-8665 • 913-851-2200
www.actuate.com/info/jpad.asp

© 2001 Actuate Corporation. Actuate is a registered trademark of Actuate Corporation. All rights reserved. Java, 100% Pure Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the U.S. and other countries. Excel is a trademark of Microsoft Corporation. All other trademarks are property of their respective owners. Specifications subject to change without notice.

Manage Documents on the Internet

Docman helps you build a document management system that looks and feels like a non-Web environment

Document management on the Internet is an important topic but is not often discussed in Java magazines. Wouldn't it be nice to have a document management system that looks and acts like your non-Web file system? Enter Docman—an experimental Web-based document management project. Docman, which is implemented using JavaServer Pages (JSP) technology, is like a Windows Explorer for the Web. It presents a Web-based hierarchical file system similar to those in non-Web environments. You can set the permission for each directory so you can control who can access what, and users can browse through the folder tree with the already familiar user interface.

> Putting BLOBs in the database is excruciatingly slow and resource hungry

This article focuses on the database design and the document download specifications of Docman. In the next issue of *Java Pro*, I will concentrate on document upload and the fully functional folder tree that looks like a Web version of Windows Explorer.

Access Problems

How do you secure a file on the Web so that only authorized people can access it? You could just put the file in one of the virtual host's subdirectories and provide no link to it, so only those who know the full path can download it. This, of course, is only as secure as the full path to the document. You probably would not secure your corporate confidential documents this way.

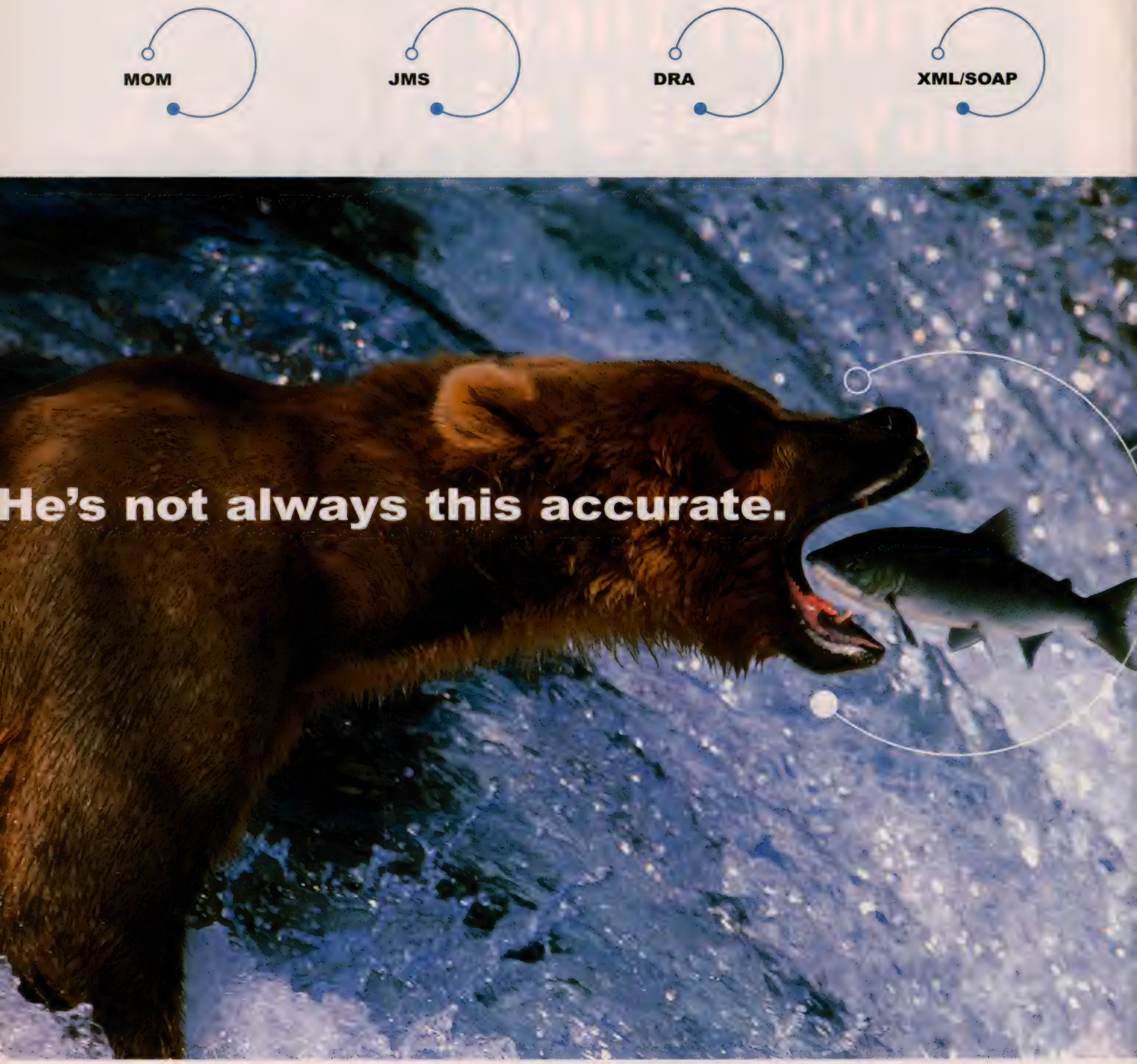
Normally, you would use the Web server and operating system authentication method so that every time someone tries to download a file, that person needs to type in the correct user name and password. This is a good approach but not always appropriate. One apparent reason why this is often not a preferred solution is that you must share the password with everyone who needs to download the file. When you share passwords, of course security will be compromised. The approach is also not suitable when you want more functionality such as document versioning and full activity records, and you don't want to rely on the Web server log files.

Another solution to securing files is to store the files in a database and allow access to them only after authenticating the user. However, putting binary large objects (BLOBs) in the database is excruciatingly slow and resource hungry. The better solution is to store those files in a directory outside the virtual host (virtual directory) so they are not visible from the Internet/intranet by typing a URL. To download a file, the user has to direct his or her browser to a special page and type in a user name and password. Upon successful authentication, the Web server fetches the file and sends it as an attachment. Using servlets or JSP, this can be achieved using this simple code:

```
if (successful user authentication) {
  // fetch the file
  response.setContentType(
    "APPLICATION/OCTET-STREAM");
  response.setHeader(
    "Content-Disposition",
```

by Budi KURNIAWAN

Budi Kurniawan is an independent consultant in Sydney, Australia, and the author of *Web Tips and Techniques*, to be published by APress this year. He is also one of the developers at BrainySoftware (www.brainysoftware.com). You can read him at **budi@brainysoftware.com**.



Java Compatible™
Enterprise Edition

©Copyright 2001 Sonic Software Corporation. All rights reserved. SonicMQ and Dynamic Routing Architecture are trademarks of Sonic Software Corporation in the U.S. and other countries. Java and

Every year the big brown bears *(Ursus arctos)* of Alaska's Kodiak Island feast—with extraordinary success—on the salmon returning to spawn in their home head waters on the Karluk River.

We are.

Worried about losing business-critical information you send over the Internet?

You should be— but not if you rely on SonicMQ™.

SonicMQ is the award-winning next-generation E-Business Messaging Server from Sonic Software. Unlike traditional message-oriented middleware products, **SonicMQ** is designed expressly for the Internet, and its patent-pending Dynamic Routing Architecture™ (DRA) incorporates durable subscriptions and persistent messaging to guarantee information delivery with absolutely no data loss. Plus, **SonicMQ** delivers the real-time performance and massive scalability required by e-business applications—while being exceptionally easy-to-use.

Find out more and learn why SonicMQ is being called the "de facto standard for JMS messaging." Visit www.sonicsoftware.com/jp.

sonic
SOFTWARE™
A Progress Software Company

ll Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other marks contained herein are the property of their respective owners.

```
    "attachment; filename=\"" +
    filename + "\"");

  java.io.FileInputStream
    fileInputStream =
    new java.io.FileInputStream(
    full file path);
  int i;
  while ((
    i=fileInputStream.read())
    != -1) {
    out.write(i);
  }
  fileInputStream.close();
  out.close();
}
```

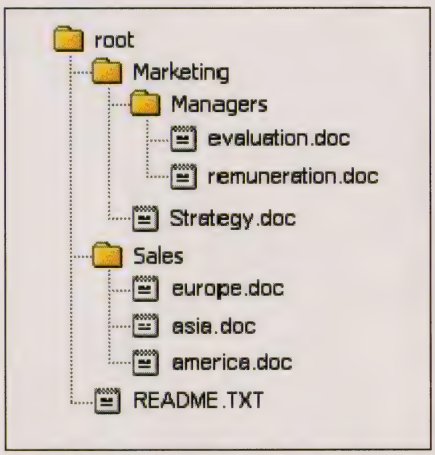


Figure 1 | Structure Your Files This figure shows the directory structure of the Docman file system in our example.

You must set the HTTP Response Content-Disposition header to attachment;filename="filename" to tell the browser to expect an attachment. The browser will then display the familiar File Download dialog.

The Docman Project

The Docman application has two types of objects: files and folders/directories. A file is any document you store in the file system. A folder or directory is a container that can contain any number of files and any number of other folders. If Folder B is located under Folder A, Folder B is said to be a subfolder of Folder A. Folder A itself is referred to as the parent folder of Folder B and any file Folder A contains. The topmost folder in the hierarchy is called the *root*. It is the first folder that the user sees when he or she browses the system. As expected, root does not have a parent folder.

Any number of users can access the files and folders. Each user has his or her own permission set, which means that you can allow a user to view files in one directory but not in the other. This is useful in an organization with different departments. For example, the marketing people can see the documents belonging to the marketing department but not IT-related files. Those working in the IT department, on the other hand, have access to these IT-related documents, but not to the HR's remuneration archive. The user can browse the system from root all the way down the hierarchy as long as he or she has permission.

In our example, file or folder information will be stored in a relational database. However, all files will be stored in one single folder outside the virtual root or virtual directory, regardless of their position in the hierarchy. This may sound strange at first, but will become logical as you go along.

It does not matter how the files are stored as long as they are not accessible directly by typing a URL on the Web browser, and as long as the information on the file system structure remains intact. In this implementation, each object (file or folder) has these properties:

- ID: a unique object ID
- Parent ID: the ID of the parent folder
- Type: the object type, whether it is a folder or a file. A folder has a type of 0 and a file of 1.
- Name: the name of the object. In other words, either a filename or a folder name.

Of special attention is the root, which is the first object in the system. It has an object ID of 1 and a parent ID of 0, which means it does not have a parent.

Object-related information is stored in a table called Objects. As you can see, by knowing the object ID and the parent ID of each object, you have enough information to understand the hierarchical structure of all folders and files.

Our application also has two other tables: Users and Permissions. The Users table stores information about users. It has three fields: ID (a unique ID identifying a particular user), UserName (the user's login name), and Password (the user's password). The Permissions table stores information on which user has permission to which files or folders. It has only two fields: ObjectID and UserID. If a user has permission to an object, this information must be present in the Permissions table. Therefore if user A, whose user ID is 7, has permission to access a folder whose object ID is 133, there must be a record in the Permissions table with UserID 7 and ObjectID 133.

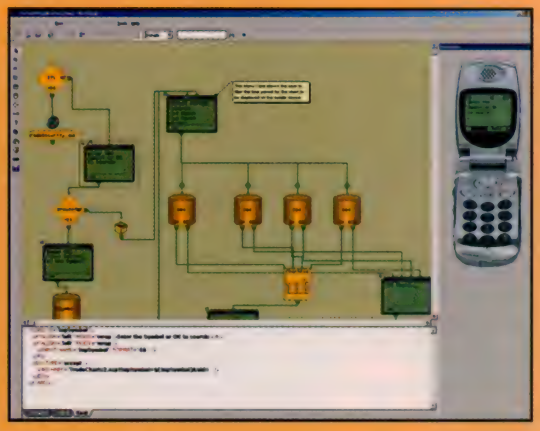
That's the theory. Now let's work on an example: the directory structure of a ficti-

Object ID	Parent ID	Type	Name
1	0	0	root
2	1	0	Marketing
3	2	1	Strategy.doc
4	1	1	README.TXT
5	1	0	Sales
6	2	0	Managers
7	6	1	evaluation.doc
8	6	1	remuneration.doc
9	5	1	europe.doc
10	5	1	asia.doc
11	5	1	america.doc

Table 1 | Table d'Object Here we see the object-related information for the file directory structure shown in Figure 1.

OPEN • EXTENSIBLE • SCALABLE

When it comes to wireless, AireLogic makes sense...



AireLogic is an **IDE** that gives a developer the ability to create wireless applications in a minimal amount of time without the need of intensive programming knowledge. With AireLogic's easy-to-use, flowchart-style interface, a developer can easily create an application's logic, bind the database connectivity and automatically generate server-side code. AireLogic fully supports connectivity with any data source.

Create the application's logic, secure the **database connectivity** and utilize the intuitive **debugging** environment, then let AireLogic do the rest. Simply select the server platform, scripting language and target device and AireLogic will automatically generate all the necessary **server-side code**, then just publish it directly to your **existing web server infrastructure,** *without any other server presence necessary.* Need to deploy your existing application to new parameters? Simply republish with the new parameters.

Whether you are wirelessly enabling an existing Internet site or legacy system, a service portal or simply extending a corporate intranet, *AireLogic just makes sense.*

AireWeb
Internet Anywhere.

AireWeb Technologies, Inc.
5200 Town Center Circle
Tower 1, Suite 450
Boca Raton, FL 33486
(866) AIREWEB toll-free
(561) 447-7038
(561) 447-8582 fax
www.aireweb.com
wap.aireweb.com

©2001 AireWeb Technologies, Inc. All rights reserved. All trademarks are registered to their respective owners.

User ID	UserName	Password
1	boni	Secret
2	bulbul	Brownhair

Table 2 | Know Your Users The Users table stores information about users. In our simple example, we have two users—one with the username "boni" and the other with the username of "bulbul."

tious company called Door Never Closed Pty Ltd. The file and folder collection is small for the sake of simplicity, yet the hierarchy reflects that of a real system. The hierarchy of all the folders and files is shown in Figure 1.

Root has two directories: Marketing and Sales. (The README.TXT is basically boring instructions from the system administrator on how to use the system.) The Marketing folder is for the Marketing department to store their folders and files. Of course, access to this folder is limited to the marketing people only. The Sales folder is for the Sales department and only the employees working in the Sales department can access it.

The content of the Objects table looks like Table 1. The ID is assigned to an object at the time of the file upload. Therefore there is no guarantee that files in a folder will have consecutive object IDs. It doesn't matter, however, because object IDs and other metadata are invisible to the user. All the user sees is a hierarchical file system. Note also that files are stored in one single directory using their object IDs. This means that the Strategy.doc file is stored as 3. Because all files are given unique numbers as names, you can have two or more files with the same filename. When a file is sent as an attachment, the system will translate the object ID into the original filename.

To keep things simple, we have only two users in this example: one with the login name "boni" and the other with the login name "bulbul." The content of the Users table looks like Table 2.

Now comes the tricky part—the permission set. Boni is the marketing manager. She has access to the marketing department's files, but has nothing to do with the sales department. Bulbul, on the other hand, is the national sales director of Door Never Closed Pty Ltd. He has access to all sales department files, but marketing is not his territory. The README.TXT is for everyone, so both Boni and Bulbul have access to it. This description is reflected in the Permissions table in Table 3.

From Database to Code

Once the database is in place, the coding part is easy. In fact, a complete system like this has only three files and one bean for database and object manipulation. The three files are:

- Login.html: a simple HTML file for users to type in their user name and password.
- DisplayObjects.jsp: a JSP file that displays all objects in a particular folder.
- DownloadObject.jsp: a JSP file that fetches a file and sends it to the browser as an attachment.

The DBBean bean does most of the work in our application. (The complete code for the DBBean is available online at www.javapro.com.) You use DBBean to connect to the database, get all objects in a particular folder, and get a file for download. Of particular importance are the two connect methods. The first method receives three parameters: url, userName, and password. The url is your JDBC URL, whereas userName and password are the connection details that you need to get access to the database. There is only one userName and one password for getting a database connection. This userName and password pair is used by the system only and is referred to in the bean as dbUserName and dbPassword. Application end users know nothing about this. Don't confuse them with the username and password

Object ID	User ID
1	1
1	2
2	1
3	1
4	1
4	2
5	2
6	1
7	1
8	1
9	2
10	2
11	2

Table 3 | You Have My Permission The Permissions table tells us which users (see the User IDs shown in Table 2) have access to which files (see Object IDs shown in Table 1).

Listing 1 Starting at the Beginning

```
<HTML>
<HEAD>
<TITLE>Login Page</TITLE>
</HEAD>

<BODY>
<H2>Door Never Closed Pty Ltd</H2>
<FORM METHOD="POST" ACTION="DisplayObjects.jsp">
<TABLE BORDER="0">
<TR>
  <TD>User Name</TD>
  <TD><INPUT TYPE="TEXT" NAME="userName"></TD>
</TR>
<TR>
  <TD>Password</TD>
  <TD><INPUT TYPE="PASSWORD" NAME="password"></TD>
</TR>
<TR>
  <TD COLSPAN="2" ALIGN="RIGHT">
    <INPUT TYPE="SUBMIT" VALUE="LOGIN">
  </TD>
</TR>
</TABLE>
</FORM>
</BODY>
</HTML>
```

Here is the code for the Login.html file. The page is shown in Figure 2.

Compuware can help. Take the mind-numbing task of scanning for memory problems and performance bottlenecks, for instance. With our DevPartner Java Edition, you can pinpoint memory-devouring code and improve performance in minutes.

- Performance profiling
- Memory profiling
- Test coverage
- Runtime multi-thread analysis
- Distributed application analysis support

Request an evaluation copy or a FREE "How Big Is IT? Memory Planning in Java" White Paper at compuware.com/numega or call 800-468-6342.

COMPUWARE®

Planning Developing Integrating Testing Implementing Managing Staffing

that belong to a user. The latter are used to get access to a particular object in the database tables.

The second connect method accepts no parameter. You use it when you want to use the default values in the bean or you have passed those three arguments in previous statements (you set the url, dbUserName, and dbPassword using the setDbUrl, setDbUserName, and setDbPassword, respectively). We do have another database related method: setJDBCDriverName(). You call this method if your driver name is other than sun.jdbc.odbc.JdbcOdbcDriver.

This example uses the JDBC-ODBC driver to access a Microsoft Access database on a Windows system. We must create a Data Source Name (DSN) called docman for the application to work. However, the choice of database is up to you. You just need to make sure that you use the correct JDBC driver. Because of the SQL statements used, your database must support subqueries.

Here is the code for the connect methods:

```
try {
  Class.forName(JDBCDriverName);
  connection =
    DriverManager.getConnection(
    url, userName, password);
}
catch (Exception e) {}
```

This could be different, depending on your database JDBC driver. Refer to the documentation for details.

The other methods used in the bean are verifyUser(String userName, String password); getDataPath(); getFolderName(String id, String userName, String password); getParentId(); getChildObjects(String parentId, String userName, String password); getFilename(String objectId, String userName, String password); and getCookie(HTTPServletRequest request, String cookie). What do each of these methods do?

You use the verifyUser(String userName, String password) method to verify that the user is an authorized user. This method checks the Users table for the UserName *username* and Password *password*. It returns true if it can find the username and password in the table, otherwise it returns false (not an authorized user). The SQL statement used is:

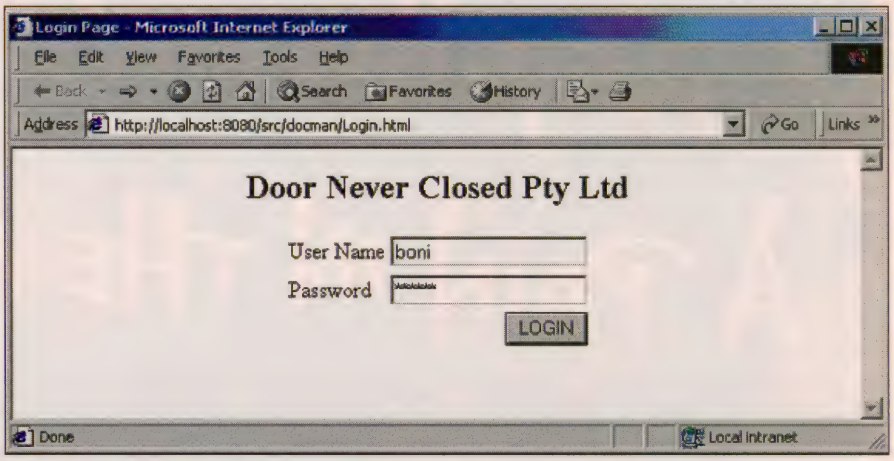


Figure 2 | The Login Page This page is a simple HTML file with a form (see Listing 1) whose ACTION attribute is set to DisplayObjects.jsp. It is the first page in our application.

```
SELECT ID FROM Users
WHERE UserName = userName
  AND Password = password
```

The getDataPath() method returns the physical directory on the server where you store all the files. You use this method when downloading or uploading a file. The data path must be set to a directory outside the virtual host.

You call the getFolderName(String id, String userName, String password) method in the DisplayObjects.jsp to get the folder name of the current folder. You need this information because you need to display the folder name to make users aware of their position in the hierarchy tree. You might think that you could easily get the folder name and parent ID by using this SQL statement:

```
SELECT Name, ParentID
FROM Objects
WHERE ID=id
```

However, we want to make sure that only users with sufficient permission can get information about specific folders. Therefore, we use this more complex SQL statement instead:

```
SELECT O.Name, O.ParentID
FROM Objects O, Permissions P,
  Users U
WHERE O.ID = id
  AND O.ID = P.ObjectID
  AND P.UserID = U.ID
  AND U.UserName = userName
  AND U.Password = password
```

The getParentId() method returns the object ID of the parent folder for the current folder. The SQL statement that retrieves a ParentID only if the user has permission to view the parent object is:

```
SELECT O.ParentID
FROM Objects O, Users U,
  Permissions P
WHERE O.ID = objectID
  AND O.ParentID = P.ObjectID
  AND P.UserID = U.ID
  AND U.UserName = userName
  AND U.Password = password
```

The getChildObjects(String parentId, String userName, String password) method returns an ArrayList containing all child objects in the current folder. We will need the ID, Type, and Name of each object. These three fields are joined together into a string and each field is separated with a comma. Again, rather than using this SQL statement:

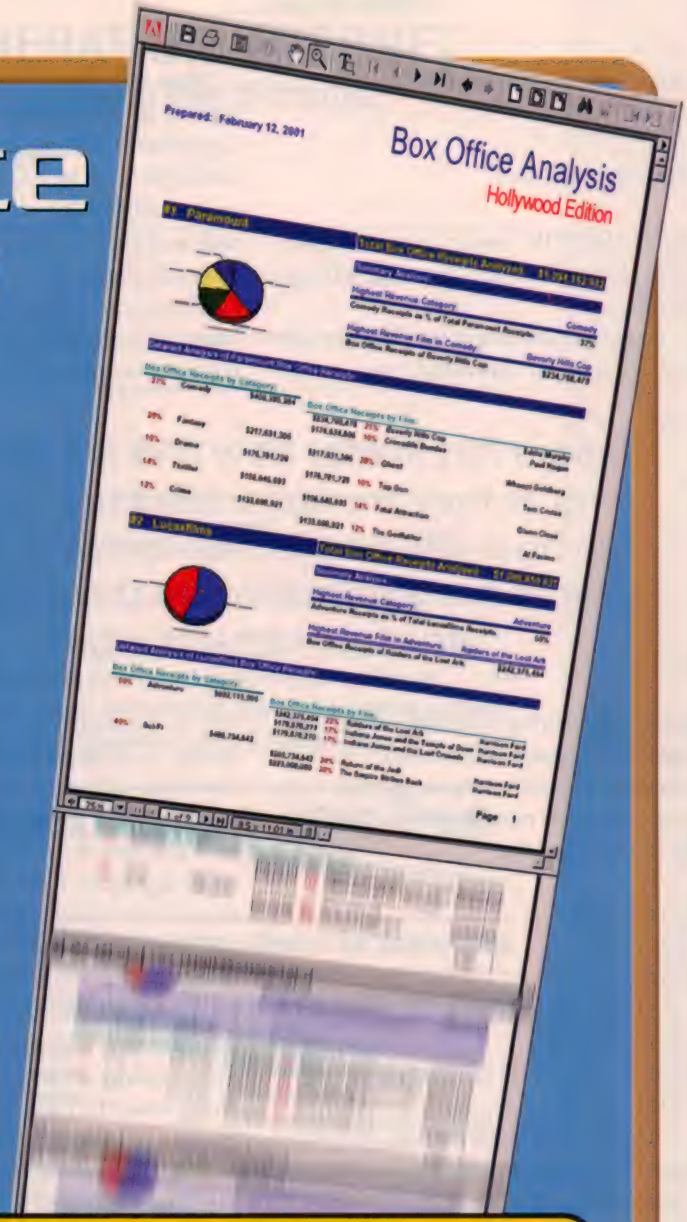
```
SELECT ID, Type, Name
FROM Objects
WHERE ParentID = id
```

We use the more "secure" SQL statement:

```
SELECT ID, Type, Name
FROM Objects
WHERE ParentID = (SELECT
  ObjectID FROM Permissions
WHERE ObjectID = id
  AND UserID =
    (SELECT ID FROM Users
WHERE UserName = userName
```

Integrate Reports

Under The Web Portal Server Of *Your* Choice



You deserve the freedom to integrate Report Servers under the Java or COM standards-based Web Portal Server(s) of your choice. Dynalivery uniquely follows this principle in building its "**Parallel Crystal**" middleware around Seagate Crystal Reports.® Since 1998, we have assisted over 100 worldwide customers to integrate Crystal Reports with Web Portal Servers for Internet reporting and publishing through Adobe® Acrobat.® If you would like to work with a partner whose focus and passion for custom and scalable reporting applications is second to none, let us know how we can help.

Visit **www.Dynalivery.com** to download FREE videos and Java samples that demonstrate Parallel Crystal integrating with these popular products.

SILVERSTREAM® | JRUN™ / COLDFUSION® | IBM® WEBSPHERE® | BEA® WEBLOGIC®

314-205-8995

888-662-7771
In North America

Dynalivery™

www.Dynalivery.com

Partner inquiries welcome

Dynalivery is a trademark of Dynalivery Corporation. Crystal Reports is a registered trademark of Crystal Decisions, Inc. All other names are trademarks or registered trademarks of their respective owners.

```
  AND Password = password
  )
)
ORDER BY Type ASC
```

The ORDER BY clause is there to guarantee that all folders (Type=0) will appear first.

To get the filename of the file object the user requests to download, we use the getFilename(String objectId, String userName, String password) method. This SQL statement is used to guarantee that only users who have permission to access this file can download it:

```
SELECT Name
FROM Objects
WHERE ID =
  (SELECT ObjectID FROM
  Permissions
WHERE ObjectID = objectId
  AND UserID =
    (SELECT ID FROM Users
WHERE UserName = userName
  AND Password = password
  )
)
```

The getCookie(HTTPServletRequest request, String cookie) method returns the value of a cookie if the cookie is found in the HTTP Request. Otherwise, it returns null. Finding a cookie is done by looping the Cookie array of the HTTP request, then comparing the cookie sought and the name of each cookie in the array.

```
Cookie cookies[] =
  request.getCookies();
if (cookies!=null)
  for (int i=0; i<
    cookies.length; ++i)
if (cookies[i].getName().equals(
  cookie))
return cookies[i].getValue();
```

Now let's examine each HTML/JSP page in our application. The first page is Login.html. This is a simple HTML file with a form whose ACTION attribute is set to DisplayObjects.jsp. The HTML code is given in Listing 1 and the page is shown in Figure 2.

Once the user types in the username and password and presses the Submit (Login) button in the Login.html, the HTTP request will be processed by DisplayObjects.jsp (listing available online). The page first checks the validation of the username and password. If the username and password are not valid, then the user is redirected to the Login page again. If they are valid, the page then tries to display the objects.

> It does not matter how files are stored as long as they are not accessible directly by typing a URL on the Web browser, and as long as the information on the file system structure remains intact

Cookies are used so that the user needs to log in only once. When a successful login happens, the user name and password are sent to the browser as cookies. These cookies are sent back to the server on the next HTTP request. The cookies live on until the user closes the browser.

The DisplayObjects.jsp is a recursive page that gets called from the link in its own body. It first tries to find the request parameters "userName" and "password."

```
String userName =
  request.getParameter(
  "userName");
String password =
  request.getParameter(
  "password");
```

This assumes that the request is from the Login page. However, the DisplayObjects.jsp can also be called from within itself. If the latter is the case, the request won't include username and password parameters. However, cookies of the same names must exist, so the code proceeds with checking for the cookies in the HTTP request.

```
if (userName==null ||
  password==null) {
  // either userName or password
  // is not found, the request is
  // not from the Login page,
  // check cookies
  userName = DBBeanId.getCookie(
    request, "userName");
  password = DBBeanId.getCookie(
    request, "password");
}
```

If the cookies are not found, the HTTP request must be coming from someone who is trying to bypass the Login page, or maybe the browser does not support cookies. In either case, the user will be redirected to the Login page.

If username and password exist as either request parameters or as cookies, it does not necessarily mean that the request is coming

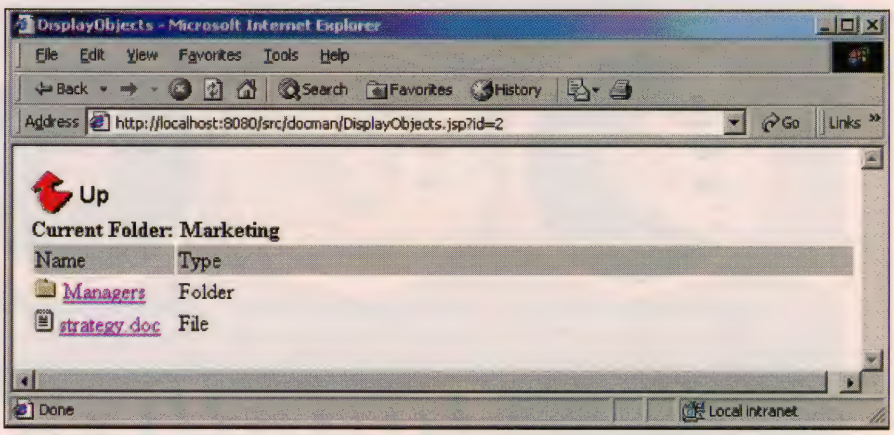


Figure 3 | Objects on Display When a user clicks on a folder, provided the user has permission to view its contents, the application will call DisplayObjects.jsp shown here, which displays all the child objects of that folder. When the user clicks on a file, provided he or she has permission to view that file, the application will send the file as an attachment to the browser.

integrated with System Definition integrated with Software Development integrated with Content Management integrated with System Testing integrated with Best Practices integrated with Change Management

System Testing integrated with Content Management integrated with Software Development integrated with Best Practices integrated with Change Management

Synchronize Your Development Team with the End-to-End Lifecycle Solution from Rational and Sun

The Rational and Sun relationship provides developers with a comprehensive lifecycle solution. Rational's developer centric tools, leadership, and Sun's platform and technology unite to give your developers a single development environment. Gain lifecycle support for J2SE™, J2EE™, and J2ME™ integrated with the Forte™ for Java™ IDE on the Solaris™ 8 Operating Environment. Maximize your technology support for building higher quality applications, faster. Put the extra power of the Rational and Sun relationship on your team and gain the flexibility, extensibility and compatibility that developers using Java technology choose first.

Rational
the e-development company

take it to the n th Sun microsystems

JDJ READERS' CHOICE AWARD

JDJ EDITOR'S CHOICE AWARD

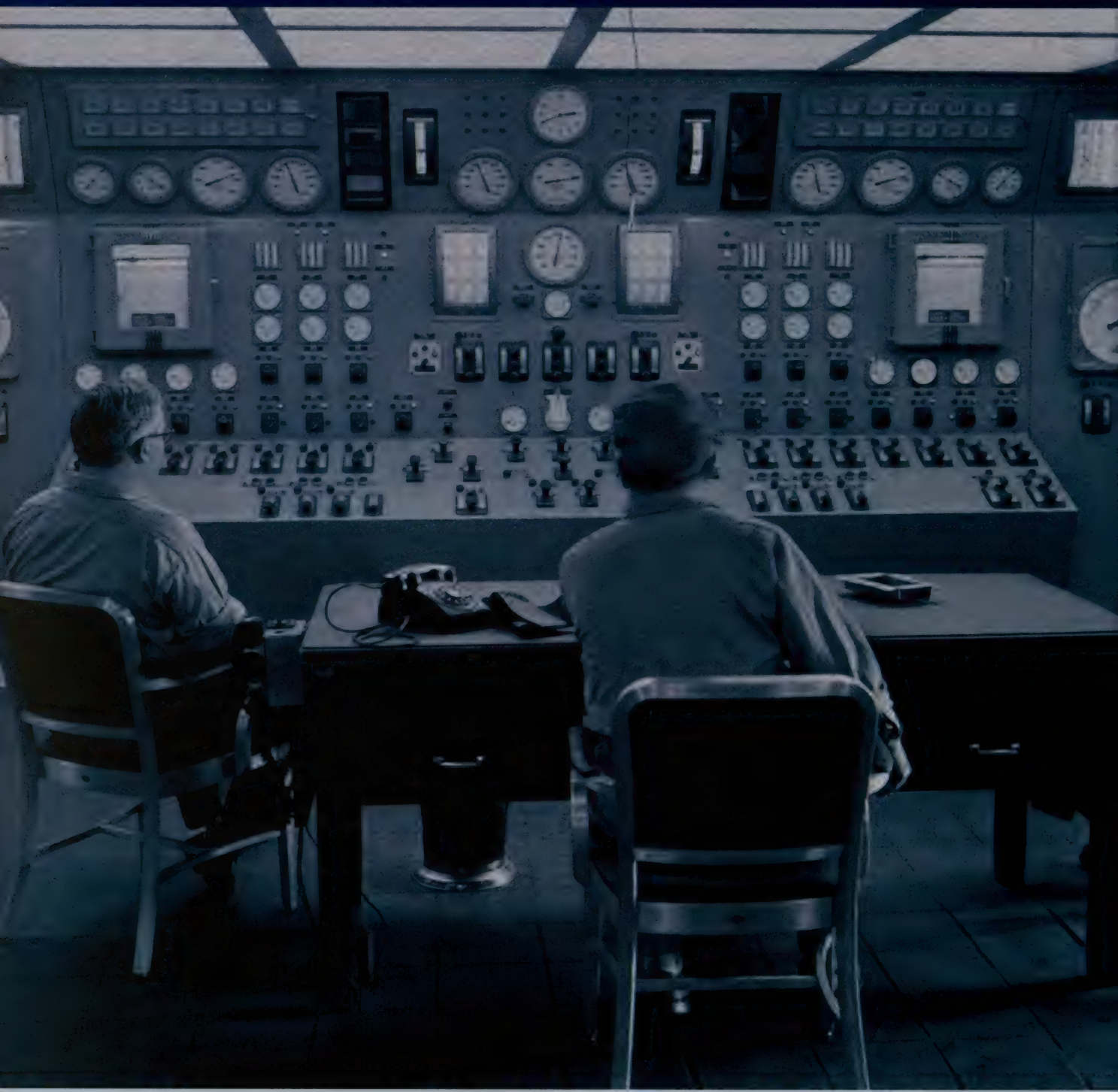
1999 JAVAPro READER'S CHOICE

JAVA REPORT 2000 READERS' CHOICE

Log-on to www.rational.com/offer/sun for more information about the Rational and Sun end-to-end lifecycle solution and a FREE Java white paper and CD.

© 2001 Rational Software Corporation. Rational, the Rational logo, Rational the e-development company, Rational Rose among others are trademarks or registered trademarks of Rational Software Corporation in the United States and/or other countries. Sun, Sun Microsystems, the Sun logo, Java, J2SE, J2EE, J2ME, Forte and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries.

You could buy our competitor's software development tools, but then you would have to use them.



www.embarcadero.com

© 2001 Embarcadero, the Embarcadero Technologies logos and all other Embarcadero Technologies product or service names are trademarks of Embarcadero Technologies, Inc. All other trademarks are property of their respec

round-trip ER/UML modeling with ER/Studio®

real-time team collaboration

integrated modeling and development environment (IMDE)

Where is it written that a program has to be hard to use just because you want to speed up your software design and collaborative development efforts? The fact is, it doesn't. Embarcadero's Describe™ gives you all the benefits of the UML (Unified Modeling Language) without having to subject your team to some complex, overpriced and totally inflexible application development tool.

That should be a huge relief. After all, your tools should eliminate complexity, not add to it. Describe simplifies analysis, design and application implementation for organizations working with Java, C++ and IDL. With features that include seamless integration with popular IDEs like Sun™ Forte™ for Java,™ real-time team collaboration, and round-trip ER/UML modeling, Describe streamlines any design, development or application re-engineering effort without forcing people to change the way they like to work.

Simpler really is better. For more information, or to download a free evaluation copy, visit **www.embarcadero.com** today.

EMBARCADERO TECHNOLOGIES®

Listing 2 Downloading Objects

```
<%@ page import="java.io.FileInputStream"
%><jsp:useBean id="DBBeanId" scope="page"
class="docman.DBBean" /><%
  // the tags constructs above are intentional
  String objectId = request.getParameter("id");
  String userName =
    DBBeanId.getCookie(request, "userName");
  String password =
    DBBeanId.getCookie(request, "password");

  if (userName==null || password==null)
    response.sendRedirect("Login.html");

  // 'userName' and 'password' cookies are found
  // now let's see if user has access to the file
  String filename = DBBeanId.getFilename(objectId,
    userName, password);

  if (filename==null)
    response.sendRedirect("Login.html");

  response.setContentType(
    "APPLICATION/OCTET-STREAM");
  response.setHeader("Content-Disposition",
    "attachment; filename=\"" + filename + "\"");

  String dataPath = DBBeanId.getDataPath();
  FileInputStream fileInputStream =
    new FileInputStream(dataPath + objectId);
  int i;
  while ((i=fileInputStream.read()) != -1)
    out.write(i);

  fileInputStream.close();
  out.close();
%>
```

Here is the code for the DownloadObject.jsp file.

from an authorized user. We still need to verify the username and password against the database every time this page is requested. If verification fails, the page sends the user back to the Login page. If it is successful, the code creates the "userName" and "password" cookies so that future requests from the user in the session will be successful. This is done here:

```
if (!DBBeanId.verifyUser(
  userName, password))
  response.sendRedirect(
  "Login.html");
else {
  // authorized user, create
  // cookies here
  Cookie cookieUserName =
    new Cookie("userName",
    userName);
  Cookie cookiePassword =
    new Cookie("password",
    password);
  response.addCookie(
    cookieUserName);
  response.addCookie(
    cookiePassword);
}
```

Every HTTP request for this page should bring with it the folder ID of the folder whose child objects are to be displayed. An exception to this is when the user first logs in. In this case, folder ID 1 (root) is implied and the atRoot flag is set. For example:

```
String id = request.getParameter(
  "id");
boolean atRoot = false;
if (id==null)
  id = "1";
// 1 is the root

if (id.equals("1"))
  atRoot = true;
```

If the current folder is not the root, the folder must have a parent folder. This code creates the link on which the user can click to go back to the parent folder:

```
if (!atRoot) {
  // display the link to go up one
  // level. This link calls this
  // same page, passing the
  // parent id
  parentId = DBBeanId.getParentId(
    id, userName, password);
  out.println("<A
    HREF=\"DisplayObjects.jsp?id="
    + parentId + "\">"
    + "<IMG SRC=\"images/
    Up.gif\" " + "BORDER=\"0\">
    </A><BR>");
}
```

Note that we use the getParentID() method of the DBBean to get the parent ID of the current folder. This parent ID is then fed as the value for the ID parameter of the DisplayObjects.jsp page. This shows the recursive nature of this page.

To give the user indication of his or her position in the hierarchy, we display the folder name here. Notice that if an unauthorized user tries to bypass the Login page, the getFolderName() method returns null.

```
String currentFolderName =
  DBBeanId.getFolderName(
  id, userName, password);
if (currentFolderName==null)
  out.println("<B>You don't have
  permission to" + " view this
  folder.</B>");
else {
  out.println(
    "<B>Current Folder: "
    + currentFolderName + "</B>");
```

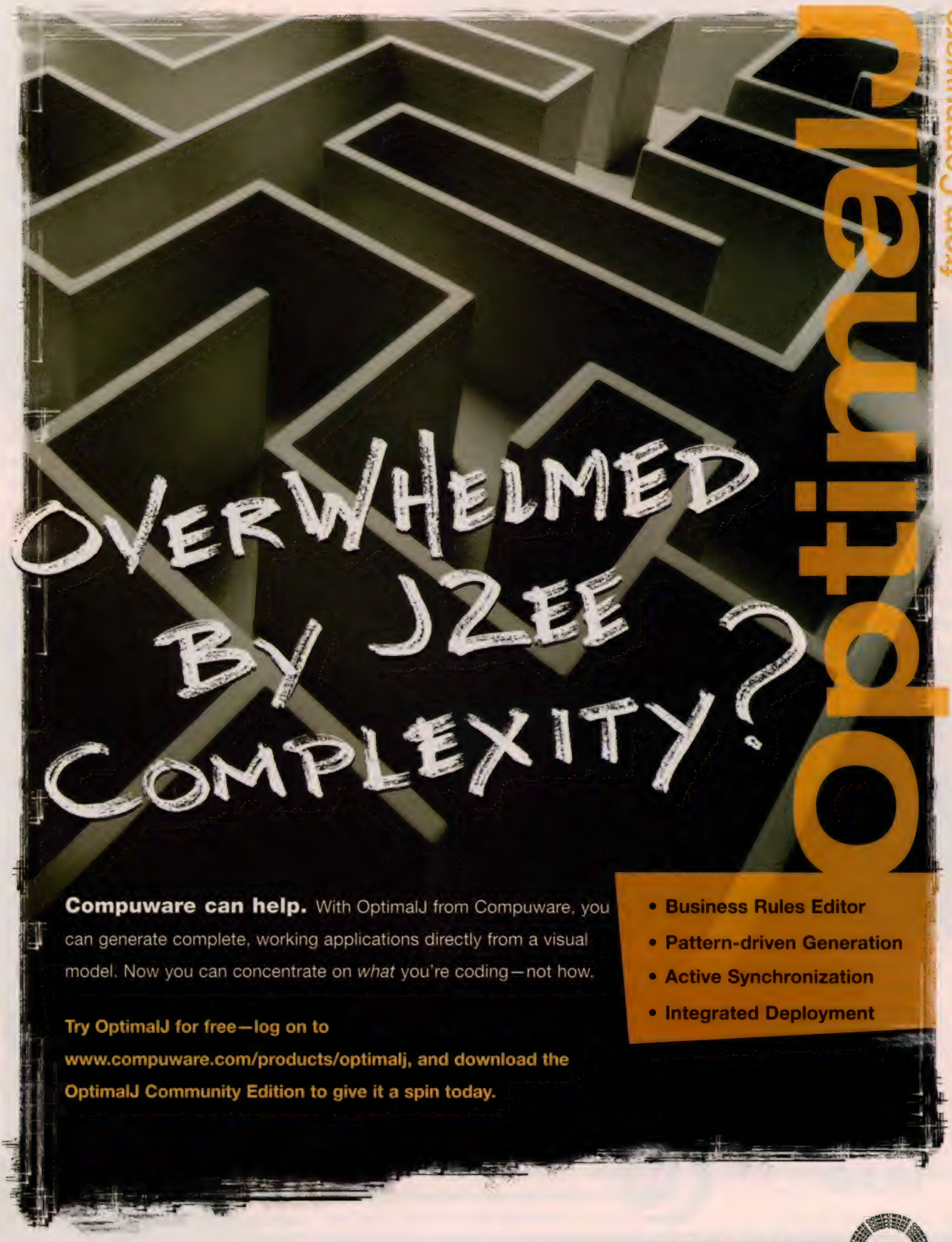
Now comes the main method that retrieves all the child objects of the current folder. The getChildObjects() method of the DBBean returns an ArrayList. Again, only authorized users can view this.

```
ArrayList records =
  DBBeanId.getChildObjects(
  id, userName, password);
```

Each item in the records ArrayList is a String containing the Object ID, Type, and Name of the child object separated by commas. Therefore we need a StringTokenizer to get each field. This code loops through the records ArrayList and gets the values of Object ID, Type, and Name of each object:

```
for (int i=0; i<
  records.size(); i++) {
  String row =
    (String) records.get(i);
  StringTokenizer st =
    new StringTokenizer(row, ",");
  String objectId = (String)
    st.nextElement();
```

Planning Developing Integrating Testing Implementing Managing Staffing



Compuware can help. With OptimalJ from Compuware, you can generate complete, working applications directly from a visual model. Now you can concentrate on *what* you're coding—not how.

- **Business Rules Editor**
- **Pattern-driven Generation**
- **Active Synchronization**
- **Integrated Deployment**

Try OptimalJ for free—log on to www.compuware.com/products/optimalj, and download the OptimalJ Community Edition to give it a spin today.

COMPUWARE®

```
  String type =
    (String) st.nextElement();
  String name =
    (String) st.nextElement();
  String imagePath;
  String objectLink;
  String objectType;
  .
  .
  .
}
```

Wanting to give users a descriptive visual interface, we use different icons to represent files and folders. To be precise, for each folder we use the Folder.gif image and for each file we use the File.gif image. The other difference is the destination links of each type. When the user clicks on a folder, the application will display all the child objects of that folder, provided the user has permission to view them. This will call DisplayObjects.jsp. When the user clicks on a file, however, the application will send the file as an attachment to the browser if the the user has permission to view that file. This procedure is depicted here:

```
if (type.equals("0")) {
  // folder
  imagePath = "<IMG SRC=
    \"images/folder.gif\"
    BORDER=\"0\">";
  objectLink = "<A HREF=
    \"DisplayObjects.jsp?id=" +
    objectId + "\">";
  objectType = "Folder";
}
else {
  // file
  imagePath = "<IMG SRC=
    \"images/file.gif\"
    BORDER=\"0\">";
  objectLink = "<A HREF=
    \"DownloadObject.jsp?id=" +
    objectId + "\">";
  objectType = "File";
}
```

Finally the child objects are displayed:

```
out.println("<TR>\n");
out.println("<TD>" +
  objectLink + imagePath +
  "</A>  " + objectLink +
    name + "</A></TD>\n");
out.println("<TD>" +
  objectType +
  "</TD>\n");
out.println("</TR>\n");
```

Figure 3 shows this page in action.

In the code for DownloadObject.jsp (see Listing 2), the first thing you'll notice is probably the construct of the first three lines:

```
<%@ page import="java.io.
  FileInputStream"
%><jsp:useBean id=
  "DBBeanId" scope="page"
  class="docman.DBBean" /><%
```

This weird construct is for the following reason. Between two JSP directives that are printed on two lines, two byte carriage returns are present between the closing tag > of the first line and the opening tag < of the second line. For example:

```
<%@ page import=
  "java.io.FileInputStream"%>
<jsp:useBean id=
  "DBBeanId" scope="page" />
```

These two extra characters are unfortunately sent to the browser with the rest of the page content. This is okay for normal HTML pages because the browser ignores white spaces, but these extra spaces are a problem when you send an attachment. The attachment that will be saved to a file must contain the exact number of the file size. Extra white spaces will corrupt the file you send, and the user may not be able to open any nontext files.

This explains the eccentric appearance of the first three lines. By putting the opening tag < of the second JSP directive right after the closing tag > of the first, we eliminate carriage return characters.

The next 17 lines in Listing 2 check the cookie to make sure that no user can bypass the Login page by typing the URL of the DownloadObject.jsp file. The last part is the code to retrieve the filename from the database and send the file as an attachment.

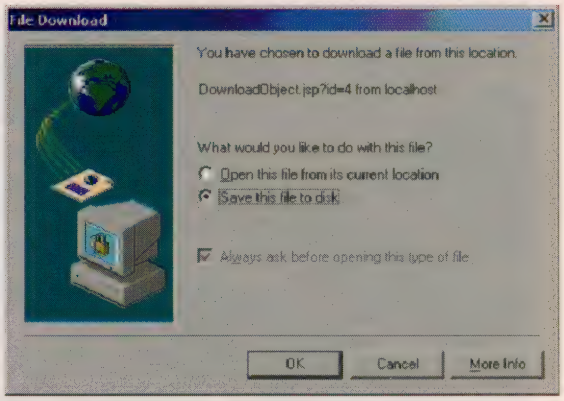

File Download
You have chosen to download a file from this location.
DownloadObject.jsp?id=4 from localhost
What would you like to do with this file?
Open this file from its current location
Save this file to disk
Always ask before opening this type of file
OK
Cancel
More Info

Figure 4 | Productive Dialog Using the getFilename() method, the actual filename of the attachment is retrieved from the DBBean. This invokes the familiar File Download dialog box shown here.

Note that the FileInputStream is instantiated by passing the object ID of the file because the filename has been changed to a unique object ID. The filename, however, is the one retrieved from the DBBean using the getFilename() method. This invokes a File Download dialog at the browser side (see Figure 4).

The Benefits

You can use Docman to manage your file system on the Web and to share documents on the Internet. You can set the permission for each folder so only authorized users can have access to certain files. The ability to download files is the first part you need to accomplish this goal. For a complete system, you also need to upload files and add records to the Permissions table for each file uploaded. This code and procedures for these operations—and a description of the JavaScript-based folder tree that will make navigation easier—can be found in the next issue. For now, check this out, and see where it takes you. JP

GO ONLINE
www.javapro.com

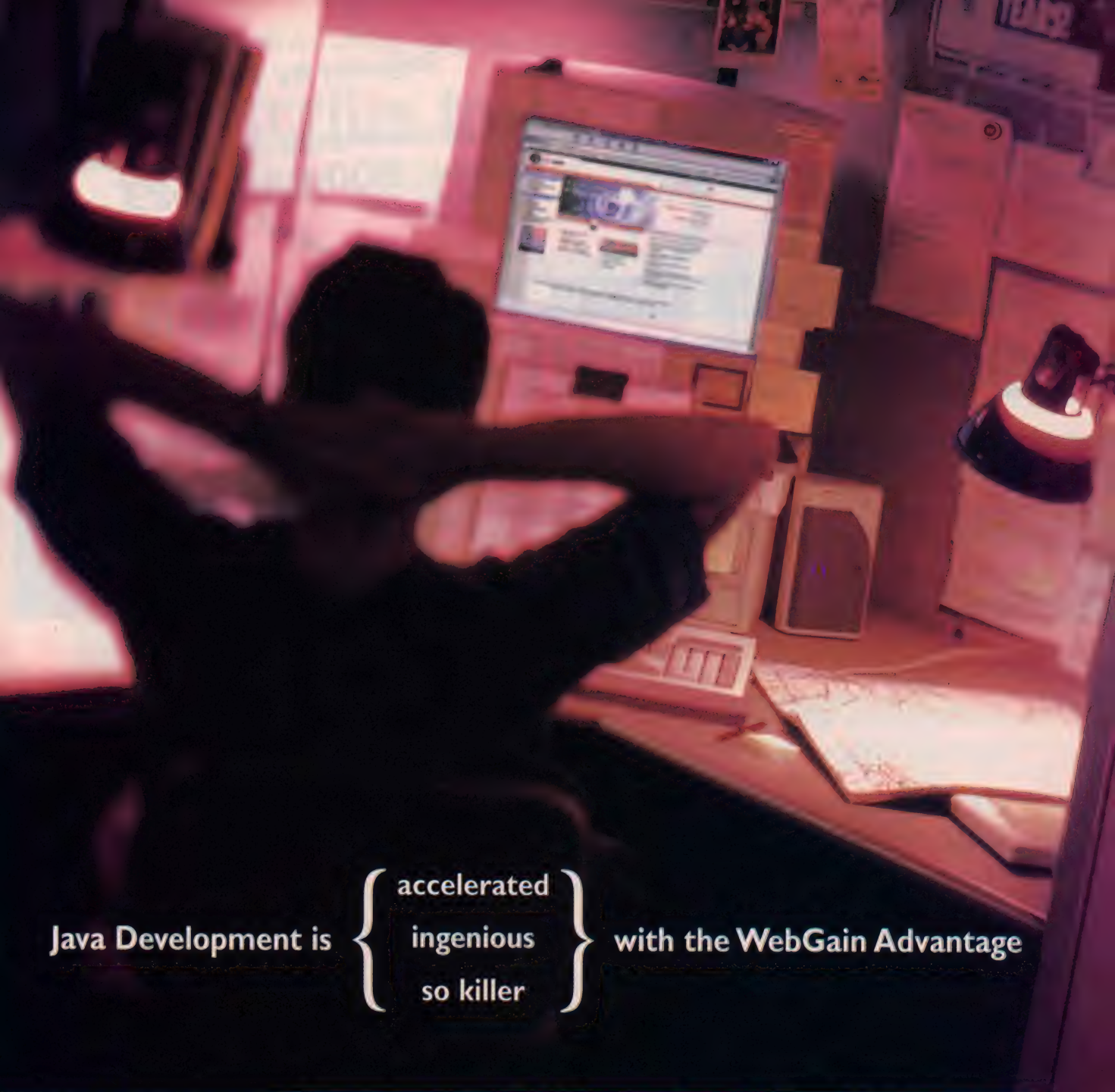
Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109BK Download the code for this article.

JP0109BK_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.



WebGain Studio™
VisualCafé™
TopLink™
StructureBuilder™
Quality Analyzer™
Application Composer™

You just deployed a super-accelerated, ingeniously built enterprise-class application on time and on spec. They said it couldn't be done. But you were smart. You gave yourself an advantage. The WebGain Advantage. The WebGain Advantage is a streamlined Java development life cycle. The key to rapidly creating and continuously improving mission critical enterprise applications. A powerfully integrated Java development solution that accelerates the process of designing, developing, deploying and integrating Java-based business applications. No wonder Fortune 5000 companies (among others) have made WebGain their choice for enterprise-level application development.

Gain the advantage by visiting www.webgain.com.

Or call us toll-free at 1.877.WEBGAIN.

Enjoy 30 days of WebGain Studio with our compliments.

webGAIN™

©2001 WebGain, Inc. WebGain, WebGain Studio, TopLink, VisualCafé, StructureBuilder, Quality Analyzer, and Application Composer are registered trademarks and service marks, or trademarks and service marks, of WebGain, Inc. in the United States and other countries.

Introducing XML MAGAZINE'S Tech Summit

Register by September 12
SAVE $200!
www.xmlmag.com/techsummit
800-848-5523

WASHINGTON, D.C.
OCTOBER 22-24

Extend Your Apps Beyond the Web with Peer and Web Services

Just a sample of what you'll get at XML Tech Summit:

- Harness UDDI, SOAP and WSDL for peer-to-peer collaboration and interoperability
- Get bleeding-edge, pragmatic advice from architects of the XML revolution and developers in the trenches already using the technology.
- Build and deploy distributed applications using XPath, XSLT, and XML Schemas
- Leverage key infrastructures from EbXML to BizTalk.
- Hear keynotes from the industry's pioneers:

Ray Ozzie
Groove Networks

Adam Bosworth
Crossgain

Call 800-848-5523 to Register!
www.xmlmag.com/techsummit

XML MAGAZINE

FTP FAWCETTE TECHNICAL PUBLICATIONS

XML Tech Summit is a trademark of Fawcette Technical Publications, Inc. *XML Magazine* is copyright © 2001 by Fawcette Technical Publications, Inc. XML is a trademark of MIT and a product of the World Wide Web Consortium. Java is a trademark of Sun Microsystems. All other trademarks are property of their respective owners.

The Zen of Xang

Apache Xang provides a fully open way of dynamically generating data-driven Web pages

by Peter VARHOL

Peter Varhol is a technical evangelist for Compuware Corporation. You can reach him at **Peter.Varhol@compuware.com**.

Although writing dynamically created Web pages based on user input for requests is a necessity today, there is no standard way of doing it. You can take the Java approach and use JavaServer Pages (JSP), or the Microsoft approach with Active Server Pages (ASP). Or you can get still more specific and use Oracle's XML technology and a proprietary Java servlet, or use the open source Simple API for XML (SAX).

But any way you cut it, you have dependencies on a specific vendor or language, or required support in other components. JSP is dependent on Java, of course, while ASP is a proprietary Microsoft technology. XML is an attractive approach because it's a widely accepted standard. But Oracle requires the use of its own Java servlet to process XML and manage the data flow, while SAX needs support in both the XML parser and the client Web browser. I'm continuing to search for an approach that places as few dependencies as possible on the system architecture. I think Xang comes the closest so far.

Xang is an Apache project that provides a way to build dynamic Web pages that integrate data from multiple sources. In conjunction with other Apache projects, it provides a full solution to creating data-driven Web pages. The good thing about Xang is that it doesn't require any proprietary technologies, or any assumptions concerning nonstandard support in the Web server or the client Web browser.

The Xang engine is implemented as a Java servlet, so you have to run it with a servlet engine. I use Apache with its Tomcat servlet engine, but any Web server and servlet engine should be satisfactory.

A Xang application is contained in a single file with the .XAP extension. A XAP file is actually an XML file that is kept on the Web server and can be called up by a client request. The file defines locations of data and data files, where to find methods for execution, and the actual logic of the methods. When a client requests data that requires a page to be dynamically created and displayed as HTML, the Xang servlet loads the XAP application and passes the client request into the application. The application itself is a hierarchy of elements. The elements have a "name" attribute that is used in matching the request from the client to the method to be called.

Interestingly, data sources are URL-addressable, which means that the XAP file can use standard URL mappings to locate information and logic needed to process client requests. Xang performs a lookup of the data using the "name" attribute on the appropriate element. The location of the lookup is based on the path defined by the URL embedded in the client request.

For example, the URL http://localhost/applications/process.xap/items/magazines/JavaPro identifies a Xang file embedded within the URL called "process.xap." Inside of that XML file, there is a name element that has an attribute called "items." Inside of that element is yet another element with an attribute by the name of "magazines." This type of relationship can be continued down the rest of the hierarchy.

How do you access the methods associated with these elements? Let's say you have a line of script such as this on the client Web page:

```
<xap onPut='doPut();
  'onPutdata='run_data();'>
```

This would access the methods specified in the URL mentioned previously. A URL such as http://localhost/applications/process.xap/?do:method=putdata is interpreted by the Xang engine as instructions to call the method identified by the onPutdata attribute of the top-level element in the XAP file (in this case, run_data()).

You define the methods in your XAP file in whatever scripting language that you choose to work with. That's the purpose of the script interpreter as a part of the configuration, so that you can execute your methods. The methods can process input, access databases, run applications or other scripts, and return data back to the client page. For example, I wrote a simple Xang method to open and get data from an XML file. The Xang engine

manages the process of taking script instructions and translating them to the URL. My method let me open and extract a specific piece of data from the file.

In general, the process of calling a particular method through the XAP interface goes like this:

1. Identify the element requested by the URL clicked on by the client. For example, it may be a text file, XML-formatted file, or even a database such as MySQL (but I haven't figured out how to get that working yet).
2. Map the HTTP request to the method referred to by the name attribute on the referenced element. This is what is done with the previous URL.
3. Ready the script referenced by the element attribute that is contained in the request.
4. Make the HTTP connection to the appropriate script so that the command can be executed.

Once you have executed your method or methods, the next step depends on what you want to do with the results. In most cases, you'd like to create a page with the results. Here's where you might want to specify a style sheet to convert the XML to a specified output for display.

The appropriate place to apply an XSL file is the output from the data source, prior to passing it back to the requesting client. And, thankfully, you need to apply a style sheet only for each type of client you are serving—probably not more than two or three. This lets you do everything the same no matter what the client is, while still customizing the output for the client. In the past, customizing output for a client meant running multiple Web sites or pages, one each for desktop browsers, small-screen displays, and mobile devices. Using an XSL style sheet definition for each type of client saves coding, computing resources, and maintenance efforts.

With XSL style sheets, the output uses can be based not only on the client that made the request but also on the kind of data the user requests. For example, perhaps the user chooses information on the items for purchase, and has a choice to receive textual data or images. Which category the user chooses might determine how the retrieved data is presented through the application of the appropriate style sheet.

Preparing for Xang

To get started with Xang, you need the following software components: a Web server such as Apache; a Java servlet engine such as Tomcat; the Apache Xang engine (found in xang.jar); the Apache XML Parser (found in xerces.jar); the Apache XSLT Processor (found in xalan.jar); and an ECMAScript Interpreter (such as fesi.zip, which is included in the Xang download). If your application calls for using XML style sheets, you'll also need to use XSLT extensions for building and running style sheets, which are contained in the files called bsf.jar and bsfengines.jar. These JAR files are both included in the ZIP file containing Xang distribution.

All of these files are included in the Xang download from the Apache Web site (www.apache.org, or go straight to the Xang page at xml.apache.org/xang). It took me a couple of hours to download, install, and configure these components onto a Windows 2000 system. It required some minor adjustments to paths and scripts before I was able to run the two sample applications included in the distribution.

Depending on what other code you may be running on your pages, you may need to include other components, such as interpreters, for other scripting languages. Unless you're using a commercial language or technology, open source or free implementations will almost certainly be available for you.

To use Xang with a Web server and servlet engine, you must include the files xang.jar, xalan.jar, xerces.jar, and fesi.zip in the servlet class path of the servlet engine. You also have to register the Xang servlet so that it can be accessed and referenced appropriately by the servlet engine. The servlet class name used by Xang is org.apache.xang.net.http.object.impl.HTTPObjectServer.

Data sources are aggregated into one unified hierarchy via XML includes and Java plug-ins that implement the standard DOM APIs. These plug-ins provide two-way access to the data via the DOM API. This provides an additional interface from which you can either create pages or process instructions from the XAP file. I haven't explored the Xang servlet in any detail yet, but because it's Java, it may be possible to write Java routines that make use of its services in interesting ways.

Xang applications are run from the client Web browser by making the XAP file a part of your Web site and storing the file on the Web server. Either load the Xang files on your Web server as you would other static pages, or map a virtual folder from your Web server to the appropriate Xang application folder. In other words, the XAP file has to be able to be served up by the Web server to a client browser.

Not a Panacea

The downside of using Xang is that it tends to require more work to create dynamic Web pages than other technologies. The instructions on the Apache Web site are weak, and I had to walk through the two examples found in the download before feeling comfortable enough to start writing Xang scripts. After a couple of weeks of experimenting, I've barely managed to scratch the surface of what Xang is capable of.

You also have to take the time and effort to point to all of your data sources from your XAP page. The URL approach for pointing to data sources means that the data sources can reside on different machines, but they have to be addressable through a URL. So your data may not be protected nearly as well as you would like.

Nevertheless, there's a good deal of potential in the Xang engine for writing dynamic Web pages. Imagine being able to aggregate data from several different files and databases without investing in any of the proprietary tools designed to do so. And if at first it seems awkward and difficult to manage, over time you'll build up a set of pages and methods that will make dynamic page development as fast as with the best of commercial tools. That is what I plan to do. JP

Use the following DevX Locator+ code at www.javapro.com to go directly to this related resource.

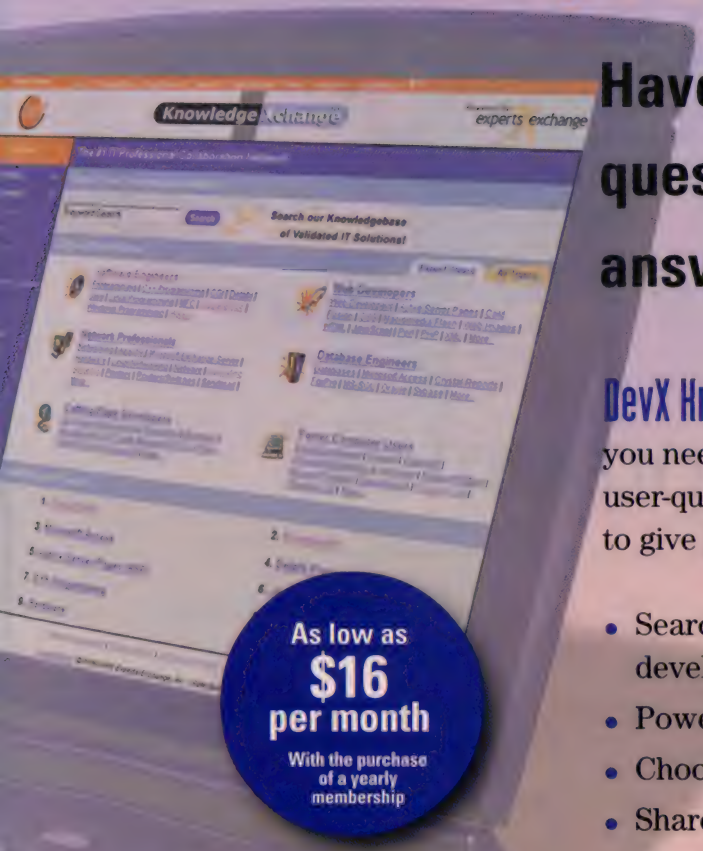
JP0109WC_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

New from DevX...

Get Answers

to Your Toughest Questions from Qualified Experts



Have a mission-critical question you need answered fast?

DevX Knowledge Xchange delivers the fast answers you need to keep your project moving by soliciting user-qualified development professionals who *compete* to give you the best solution.

- Search over 2 million timely, real-world IT development solutions
- Power search our vast library of answered questions
- Choose the best solution among qualified answers
- Share your knowledge with your peers
- Easily manage your personalized content!

Knowledge Exchange

Sign up now!

Go to DevX Knowledge Xchange at:

www.devx.com/knowledgexchange

devX

www.devx.com

DevX is a trademark of DevX.com, Inc. • 913 Emerson St., Palo Alto, CA 94301 • P: 650.566.2000

Upgrade NOW

Premier Club Delivers:

- Savings on software
- Access to the DevX development article and code archive
- Live member-only events

Join the Premier Club for $3.95 per month and your first 30 days are free!

From the DevX home page click on JOIN

Need Development Know-How?

- Tap into 1000's of programming tips
- Ask the experts
- Browse tons of content and code from top IT publications
- Join live developer forums
- Access over 2,000 product reviews
- Purchase and download leading development tools
- **Join Premier Club** for exclusive benefits

Your search stops at DevX.

Get the unbiased information and tools you need NOW from the leading information resource for IT professionals developing business applications. We connect developers with objective, technical information and tools to get the job done—code samples, technical articles, discussions, learning services, and other resources aimed at helping you solve your toughest development challenges.

To learn more, visit:

www.devx.com

DevX is a trademark of DevX.com, Inc.

devx
development exchange

A Tokenizer for Your Collection

Parsing data causing you headaches? Save time and energy when writing parsers by using this flexible, extensible, and reusable tokenizer

by Claude **DUGUAY**

Claude Duguay is vice president of engineering at 360 Powered Corp. He has developed software for more than 20 years. He is passionate about Java, reads about it incessantly, and writes about it as often as possible. You can reach him at **claude@360.com**.

Numerous applications require parsing data in one form or another. Sadly, the StringTokenizer and StreamTokenizer classes, provided as part of the standard Java distribution, fall short of what's required in most cases. Building your own tokenizer each time you need to parse something is both time consuming and usually unnecessary. In this article, I'll show you how to create a flexible, extensible tokenizer that you can reuse whenever you need to write a parser.

The FlexTokenizer uses a pluggable Tokenizer/Token paradigm. A Token represents the element that was tokenized, which could be a number, string, symbol, or other token. A Tokenizer is the engine that recognizes a Token in the sequence of text presented to it. It is responsible for recognizing and creating a given token. The FlexTokenizer allows you to add and remove Tokens and Tokenizers as you see fit.

> FlexTokenizer can be used to tokenize virtually any text source and new tokens can be created easily to handle any kind of special need

FlexTokenizer also lets you define the sequence in which tokens will be considered. For example, a "/" character may be the start of a comment or a symbol, so the comment should be considered first. If the associated Tokenizer fails, then we assume it was a symbol. The comment example is especially relevant. In our default implementation, we can tokenize both kinds of Java comments. Three kinds of Tokens can begin with the "/" character—the two comment types and a symbol token. Each needs to be considered in proper sequence.

This article was inspired by a recent book from Steve Metsker called *Building Parsers with Java* (Addison-Wesley, 2001). Although the code was written from scratch, there is no question it was heavily influenced by the ideas in this book. I recommend it highly to anyone interested in parser development. The FlexTokenizer extends these ideas in several ways.

The notion of using sequences is entirely mine, as is the inclusion of positional information with each token. In my experience, this information is indispensable when trying to debug a parser or for providing useful feedback to the user. I also make heavy use of a BufferedReader for look-aheads rather than a PushbackInputStream, which can deal with only one character at a time and deals with text at the byte level rather than the character level.

Let's take a look at the classes required to make this work. Figure 1 shows the relationships between each of the classes involved. This design has two major interfaces, Token and Tokenizer. Each interface includes default implementations as inner classes. Both of these interfaces include an Abstract implementation and the Tokenizer includes two additional abstract classes that pull together commonalties for a few of the concrete classes.

The FlexTokenizer uses Sequence objects to collect Tokenizers and associate them with initial characters. You'll see how this works when we take

a closer look at the code. The TokenReader uses the Offset class to collect position information. You'll see that each class maximizes flexibility and offers options on which you can capitalize when you have more specific needs. As it stands, this tokenizer will easily deal with Java source code. The main() method in FlexTokenizer demonstrates this capability by tokenizing its own source code.

Naturally, we don't have room to look at each of the classes in this project. You can find them all online at www.javapro.com. We'll take a look at the more important classes, including TokenReader and FlexTokenizer itself, as well as an illustrative Tokenizer and associated Token. Let's take a quick look at the two interfaces we'll be using:

```
public interface Token
{
  public Object getToken();
  public void setToken(
    Object token);
  public Offset getOffset();
  public void setOffset(
    Offset pos);
  public boolean isIgnorable();
  public void setIgnorable(
    boolean ignorable);
}
```

The Token interface provides access to the three elements included in each of our tokens. We need access to the token itself, which may be any object, such as a Character, Number, or String. We need access to the Offset object, which encapsulates position information, including an offset from the beginning of the stream, as well as the line number and line offset for the token.

The ignorable flag is used to skip over certain tokens when processing a given stream. Comments and white space, for example, can be returned or ignored by using this mechanism. Tokens that are ignorable are flagged in advance so that the FlexTokenizer can recognize ignorable tokens as it operates.

```
public interface Tokenizer
{
  public Token
    nextToken(TokenReader reader)
    throws IOException;
}
```

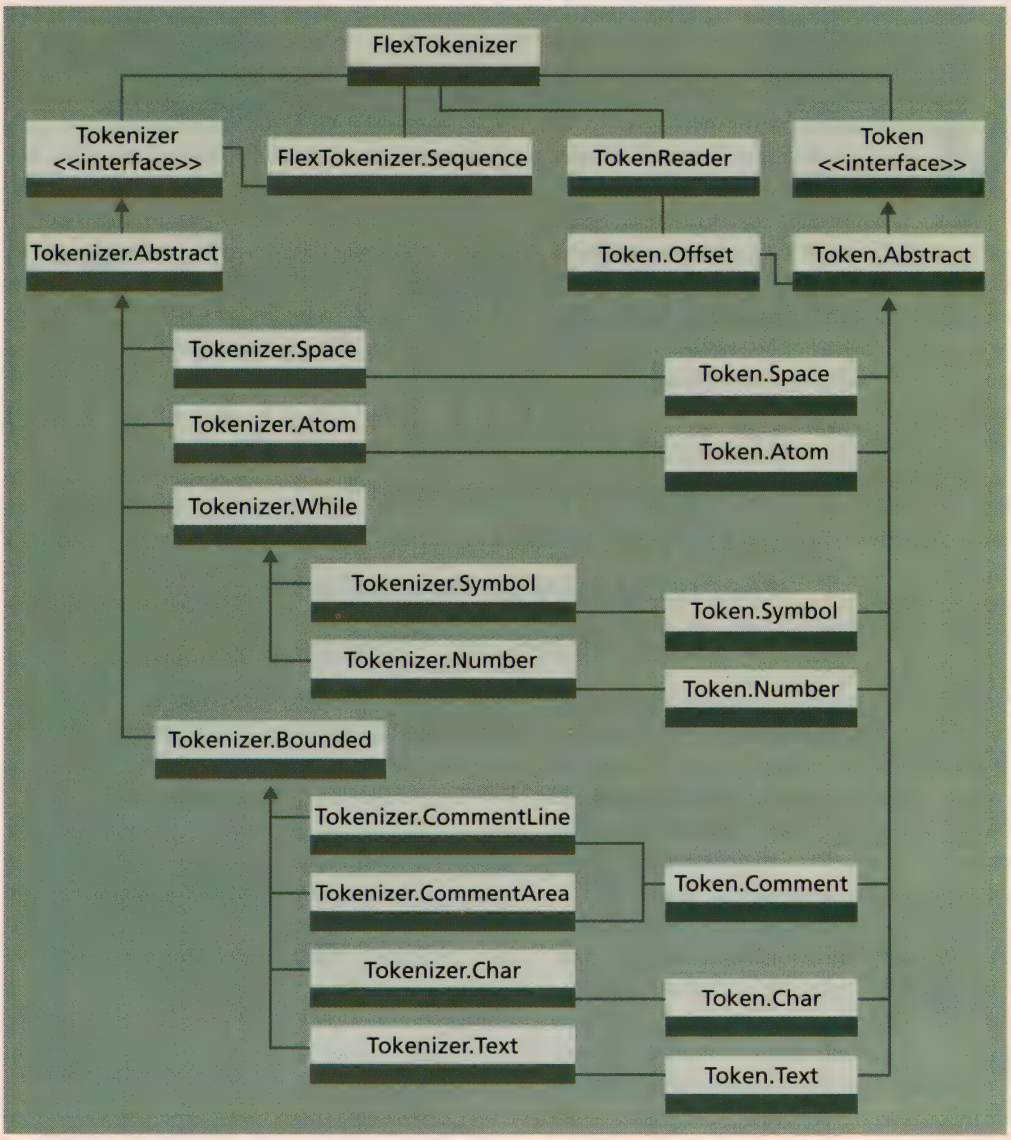


Figure 1 | FlexTokenizer Here we see the relationships between each of the classes involved in making the FlexTokenizer work.

The Tokenizer interface is simple. It takes a TokenReader reference and returns a Token object, if one can be constructed at the current position. If the Token can't be constructed, the Tokenizer returns a null value. The FlexTokenizer then resets the stream position and tries to process another Tokenizer in the Sequence, if one is available.

Before we look at the Token and Tokenizer implementation, let's take a quick look at the code for TokenReader in Listing 1. TokenReader extends BufferedReader and takes advantage of the mark() method supported by the reader. We override the read(), mark(), and reset() methods to capture position information. Because the FlexTokenizer always reads one character at a time, we can ignore the other read methods supported by the Reader interface.

All we need to do is increment our position counters every time the read() method is called. When we mark a position, the current position is kept in a set of instance variables whose names are prefixed by the word mark. When the reset() method is called, we reset the current position from these values. Notice that the super-

IBM, WebSphere and the e-business logo are registered trademarks of International Business Machines Corporation in the United States and/or other countries. ©2001 IBM Corporation. All rights reserved.

THE CODERNAUTS FIND AN ENVIRONMENT THAT SUPPORTS 35 PLATFORMS. HAIL, OPEN WORLD!

WEBSPHERE® *FOR* MANKIND

THE FASTEST GROWING E-BUSINESS SOFTWARE PLATFORM ON THE PLANET

e business software **ibm.com**/websphere/mankind

IT'S A DIFFERENT KIND *OF* WORLD.
YOU NEED A DIFFERENT KIND *OF* SOFTWARE.

class methods are called in each of these cases, so the fundamental behavior is identical to a BufferedReader.

The only other method in TokenReader is getOffset(), which constructs an Offset object from the file offset, line number, and line offset. This is used by each of the Tokenizer implementations to add offset information to the Token being constructed. We'll skip over the Offset implementation, which does little more than provide accessors for the three position integers and a toString() method to make it easy to print results.

Because the Symbol tokenizer is the one you are most likely to directly access, we'll take a quick look at the implementation. The Token.Symbol class extends Token.Abstract and exposes a constructor and a toString() method. Token.Abstract exposes the Token interface and adds a toString() method that expects two String arguments that identify the name of the token and a string representation for the value. This makes it easier for subclasses to implement their own toString() methods.

The Tokenizer.Symbol class extends Tokenizer.While, which extends Tokenizer.Abstract. The Abstract class implements three utility methods that can be used by subclasses: peek(), startsWith(), and endsWith(). The peek() method uses the TokenReader mark to mark the current position, reads a character, and resets the current position. The startsWith() and endsWith() methods operate on StringBuffer objects, using the substring() method to minimize String object creation.

The Tokenizer.While class aggregates behavior common to tokenizers that need to append characters to their stack until a mismatch is found. This is true of numbers and symbols in our implementation. Having an abstract class like Tokenizer.While helps eliminate duplication and makes it easier to create new tokenizers with the same fundamental behavior.

Tokenizer.While implements the nextToken() method required by the Tokenizer interface and uses two abstract methods. The first, isValid(), is used to validate the token buffer as we go along. When this method returns false, we have stepped past the last character in the token. The createToken() method is used when we've decided that a valid token is available. Both these method implementations are specific to a given Tokenizer subclass, though the commonality is captured by the nextToken() method.

Listing 2 shows the code for Tokenizer.Symbol, which extends the Tokenizer.While abstract class. As such, we implement the isValid() and createToken() methods. The latter is simple in that all it does is create an instance of a Token.Symbol with suitable arguments. The isValid() method always returns true for a single character and looks at an internal list to determine whether the token is currently valid.

To support multicharacter symbols, we use an extendedSymbolList instance of ArrayList, which holds a list of strings that are kept in length order. The add() method intentionally keeps longer strings near the beginning of the list so that the isValid() method does not report failure on a shorter token prematurely. Along with the add() method, we also support remove() and clear() methods to manage the symbol token list.

You'll notice that the Tokenizer.Symbol constructor sets up a series of symbols that are applicable to parsing Java. These can be cleared easily by calling the clear() method, and new symbols can be added. This is an operation you'll find fairly common when dealing with different parsers and languages. Let's take a look at the FlexTokenizer implementation to see how you can use Tokenizers to accomplish different goals.

Listing 3 shows the code for FlexTokenizer. Because I've used a lot of inner classes in this project, I've left comments in the listings indicating some removal. The

Continued on page 75.

Listing 1 Token Class

```
import java.io.*;

public class TokenReader
  extends BufferedReader
{
  protected int fileOffset = 0;
  protected int lineNumber = 0;
  protected int lineOffset = 0;
  protected int markFileOffset = 0;
  protected int markLineNumber = 0;
  protected int markLineOffset = 0;

  public TokenReader(Reader reader)
  {
    super(reader);
  }

  public int read() throws IOException
  {
    int chr = super.read();
    fileOffset++;
    lineOffset++;
    if (chr == '\n')
    {
      lineNumber++;
      lineOffset = 0;
    }
    return chr;
  }

  public void mark(int max) throws IOException
  {
    markFileOffset = fileOffset;
    markLineNumber = lineNumber;
    markLineOffset = lineOffset;
    super.mark(max);
  }

  public void reset() throws IOException
  {
    fileOffset = markFileOffset;
    lineNumber = markLineNumber;
    lineOffset = markLineOffset;
    super.reset();
  }

  public Token.Offset getOffset()
  {
    return new Token.Offset(
      fileOffset, lineNumber, lineOffset);
  }
}
```

This listing shows the code for the TokenReader class, which extends BufferedReader. This class overrides the read(), mark(), and reset() methods to capture position information.

IBM, the e-business logo, and DB2 are registered trademarks of International Business Machines Corporation in the United States and/or other countries. UNIX is a registered trademark of The Open Group in the United States and other countries. Other company, product and service names may be trademarks or service marks of others. © 2001 IBM Corp. All rights reserved.

TWO CODERNAUTS. ONE DISTRIBUTED DATABASE. THE PARALLEL UNIVERSE WAS SAVED.

DB2® *FOR* UNIX®

GROWING MORE THAN 2.5 TIMES FASTER THAN ORACLE

e business software **ibm.com**/db2/faster

IT'S A DIFFERENT KIND *OF* WORLD.
YOU NEED A DIFFERENT KIND *OF* SOFTWARE.

Listing 2 Extended Class

```
public interface Tokenizer
{
  // ... CODE REMOVED FOR LISTING ...

  public static class Symbol extends While
  {
    protected ArrayList extendedSymbolList =
      new ArrayList();

    public Symbol()
    {
      add("==");
      add("!=");
      add(">=");
      add("<=");
      add("&&");
      add("||");
      add("--");
      add("++");
      add("-=");
      add("+=");
      add("*=");
      add("/=");
    }

    public void add(String symbol)
    {
      boolean added = false;
      int len = symbol.length();
      int count = extendedSymbolList.size();
      for (int i = 0; i < count; i++)
      {
        String test = (String)
          extendedSymbolList.get(i);
        if (len > test.length())
        {
          extendedSymbolList.add(i, symbol);
          added = true;
        }
      }
      if (!added) extendedSymbolList.add(symbol);
    }

    public void remove(String symbol)
    {
      extendedSymbolList.remove(symbol);
    }

    public void clear()
    {
      extendedSymbolList.clear();
    }

    public boolean isValid(String text)
    {
      if (text.length() == 1) return true;
      int count = extendedSymbolList.size();
      for (int i = 0; i < count; i++)
      {
        String symbol = (String)
          extendedSymbolList.get(i);
        if (symbol.startsWith(text)) return true;
      }
      return false;
    }

    public Token createToken(
      Object token, Token.Offset offset)
    {
      return new Token.Symbol(
        (String)token, offset);
    }
  }

  // ... CODE REMOVED FOR LISTING ...
}
```

The code for Tokenizer.Symbol, shown here, extends the Tokenizer.While abstract class.

Listing 3 The Final Product

```
import java.io.*;
import java.util.*;

public class FlexTokenizer
{
  // ... CODE REMOVED FOR LISTING ...

  protected boolean skipIgnorables = true;
  protected int readAheadLimit = 255;
  protected TokenReader reader;

  protected Sequence[] charTable =
    new Sequence[256];

  public static final Tokenizer
    SYMBOL_TOKENIZER = new Tokenizer.Symbol();
  public static final Tokenizer
    SPACE_TOKENIZER = new Tokenizer.Space();
  public static final Tokenizer
    TEXT_TOKENIZER = new Tokenizer.Text();
  public static final Tokenizer
    CHAR_TOKENIZER = new Tokenizer.Char();
  public static final Tokenizer
    NUMBER_TOKENIZER = new Tokenizer.Number();
  public static final Tokenizer
    ATOM_TOKENIZER = new Tokenizer.Atom();
  public static final Tokenizer
    COMMENT_LINE_TOKENIZER =
      new Tokenizer.CommentLine();
  public static final Tokenizer
    COMMENT_AREA_TOKENIZER =
      new Tokenizer.CommentArea();

  public static final Sequence
    SYMBOL_SEQUENCE = new Sequence(
      SYMBOL_TOKENIZER);
  public static final Sequence
    SPACE_SEQUENCE = new Sequence(
      SPACE_TOKENIZER,
      SYMBOL_TOKENIZER);
  public static final Sequence
    TEXT_SEQUENCE = new Sequence(
      TEXT_TOKENIZER,
      SYMBOL_TOKENIZER);
  public static final Sequence
    CHAR_SEQUENCE = new Sequence(
      CHAR_TOKENIZER,
      SYMBOL_TOKENIZER);
  public static final Sequence
    NUMBER_SEQUENCE = new Sequence(
      NUMBER_TOKENIZER,
      SYMBOL_TOKENIZER);
  public static final Sequence
    ATOM_SEQUENCE = new Sequence(
      ATOM_TOKENIZER,
      SYMBOL_TOKENIZER);
```

Continued on page 75.

The code for the FlexTokenizer class, shown here, is missing the inner class Sequence, which extends ArrayList and provides a few utility constructors to help create single, dual, and triad collections of Tokenizers.

Continued from page 72.

missing inner class in the FlexTokenizer listing is for Sequence, which extends ArrayList and provides a few utility constructors to help create single, dual, and triad collections of Tokenizers. The only implemented method is getTokenizer(), which casts the lookup of a given indexed value to a Tokenizer object for convenience.

The code for FlexTokenizer relies on a character table that associates Tokenizers with characters. This mechanism allows the FlexTokenizer to use the correct Tokenizer when it encounters a given initial character. For example, a double quote character is associated with the Tokenizer.Text object. These associations are indirect in that Tokenizer objects are made part of a Sequence collection, which is then associated with a character in the charTable array.

The Sequence paradigm allows us to associate multiple Tokenizer objects with a given character in the charTable. That means that, should one tokenizer fail, with the nextToken() method returning a null value, the next Tokenizer in the Sequence can be tried. The order in which tokenizers are applied is, therefore, completely under your control.

As you look through the code, you'll see instance variables for skipIgnorables, which allows us to skip ignorable tokens in the output; readAheadLimit, which sets the maximum number of characters we can look ahead before resetting a mark in the TokenReader; the charTable, which holds character/sequence associations; and a reference to a TokenReader.

What follows is a set of declarations for Tokenizer and Sequence instances that are used by default. We do this to avoid duplication and to provide a convenient way for you to associate a combination of your

Continued from page 74.

```
  public static final Sequence
    COMMENT_SEQUENCE = new Sequence(
      COMMENT_LINE_TOKENIZER,
      COMMENT_AREA_TOKENIZER,
      SYMBOL_TOKENIZER);

  public FlexTokenizer(String text)
  {
    this(new TokenReader(
      new StringReader(text)));
  }

  public FlexTokenizer(TokenReader reader)
  {
    this.reader = reader;
    initCharTable();
  }

  public void initCharTable()
  {
    setTokenizerSequence(0, 255,
      SYMBOL_SEQUENCE);
    setTokenizerSequence(0, ' ',
      SPACE_SEQUENCE);
    setTokenizerSequence('a', 'z',
      ATOM_SEQUENCE);
    setTokenizerSequence('A', 'Z',
      ATOM_SEQUENCE);
    setTokenizerSequence(0xc0, 0xff,
      ATOM_SEQUENCE);
    setTokenizerSequence('\'', '\'',
      CHAR_SEQUENCE);
    setTokenizerSequence('"', '"',
      TEXT_SEQUENCE);
    setTokenizerSequence('-', '-',
      NUMBER_SEQUENCE);
    setTokenizerSequence('.', '.',
      NUMBER_SEQUENCE);
    setTokenizerSequence('0', '9',
      NUMBER_SEQUENCE);
    setTokenizerSequence('/', '/',
      COMMENT_SEQUENCE);
  }

  public void setTokenizerSequence(
    int from, int to, Sequence sequence)
  {
    for (int i = from; i <= to; i++)
    {
      charTable[i] = sequence;
    }
  }

  public void setSkipIgnorables(
    boolean skipIgnorables)
  {
    this.skipIgnorables = skipIgnorables;
  }

  public boolean getSkipIgnorables()
  {
    return skipIgnorables;
  }

  protected int peek()
    throws IOException
  {
    reader.mark(1);
    int chr = reader.read();
    reader.reset();
    return chr;
  }

  public Token nextToken()
    throws IOException
  {
    int chr = peek();
    if (chr == -1) return null;

    Token token = null;
    Sequence sequence = charTable[chr];
    reader.mark(readAheadLimit);
    for (int i = 0; i < sequence.size(); i++)
    {
      reader.reset();
      Tokenizer tokenizer =
        sequence.getTokenizer(i);
      token = tokenizer.nextToken(reader);
      if (token != null) break;
    }
    if (skipIgnorables && token.isIgnorable())
    {
      token = nextToken();
    }
    return token;
  }

  public static void main(String[] args)
    throws IOException
  {
    String filename = "FlexTokenizer.java";
    FlexTokenizer tokenizer = new FlexTokenizer(
      new TokenReader(new FileReader(filename)));

    Token tok;
    while ((tok = tokenizer.nextToken()) != null)
    {
      System.out.println(tok);
    }
  }
}
```

choosing. The initCharTable() method does the actual default association work. You can, therefore, either call the setTokenizerSequence() method directly to associate a Sequence with a range of characters, or you can override the initCharTable() method in a subclass.

The two constructors we provide allow you to pass in a text string or a TokenReader. We provide accessors for the skipIgnorable option and a utility method called peek() for looking ahead one character at a time. The real work is accomplished in the nextToken() method, which you can call repeatedly until you get a null Token.

After peeking at the first available character, we can find the associated Sequence in the charTable. With a Sequence in hand, we step through each Tokenizer and call its nextToken() method until we get a non-null response, which will always be the next matchable token. Before returning the token, we check whether it's ignorable and recursively call nextToken() if it is, in order to get the next available token.

I've left the main() method in the FlexTokenizer class to demonstrate typical usage. Because we can count on having access to this source code, a FlexTokenizer instance is initialized to parse the "FlexTokenizer.java" file. To do this, we wrap a TokenReader around a FileReader. The rest of the code simply calls nextToken() until we get a null value, and prints each token as we go. You'll see something suitably readable because each Token implements the toString() method.

FlexTokenizer is a powerful tokenizer, moving well beyond the StringTokenizer and StreamTokenizer classes offered with standard Java releases. FlexTokenizer can be used to tokenize virtually any text source and new tokens can be created easily to handle any kind of special need. The ability to associate sequences of tokenizers with starting characters makes it easy to build tokenizers for a wide variety of applications.

I think you'll find the FlexTokenizer architecture especially flexible and the position offset information indispensable when you develop your own parser solutions. Few existing tokenizer implementations deal effectively with escape characters in string or character literals. FlexTokenizer easily handles these issues and makes the process of implementing tokenizers virtually transparent. All these features make FlexTokenizer ultimately reusable. I hope you'll agree. JP

GO ONLINE
www.javapro.com

Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

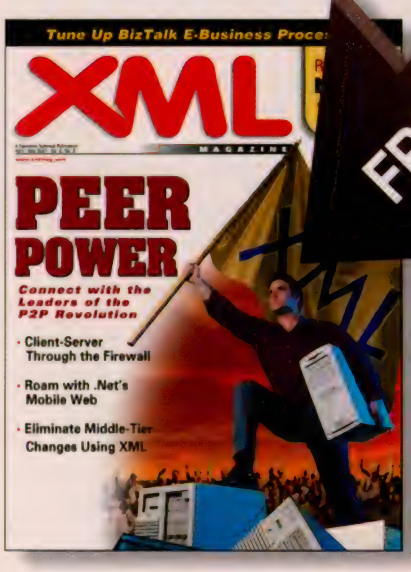
JP0109 Download all of the code for this issue of *Java Pro*.

JP0109CD Download the code for this article.

JP0109CD_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

XML Solutions and Strategies:


Tune Up BizTalk E-Business Proce
XML
MAGAZINE
PEER POWER
Connect with the Leaders of the P2P Revolution
· Client-Server Through the Firewall
· Roam with .Net's Mobile Web
· Eliminate Middle-Tier Changes Using XML

FREE Subscription!

XML Magazine is the only independent publication devoted 100-percent to the latest information about using XML for enterprise-level challenges.

Published every other month, you'll get practical, hands-on articles, tips and enterprise-level solutions, including:

- Pragmatic techniques for developing with XML
- Complete source code examples
- Independent product reviews
- Columns by the industry experts who design the technology
- Cutting-edge analysis of industry trends

Sign up today for your FREE subscription @ www.xmlmag.com/free

Get them here first!

Publisher reserves the right to determine qualifications for free subscriptions. XML is a trademark of MIT. *XML Magazine* is published by FTP, Inc.

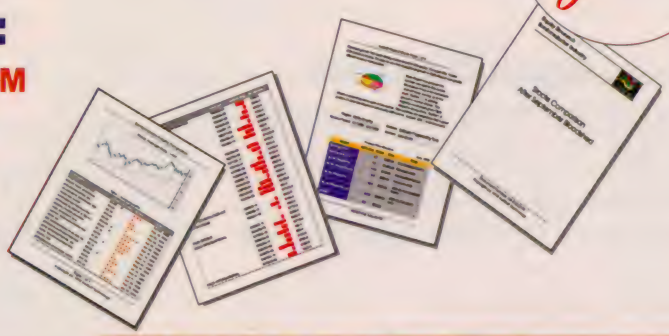
Reporting the hard way:

SUPPORTS Adobe Acrobat

Reporting the easy way:

Style Report™

Java printing, Web reporting



Style Report family provides highly flexible printing/reporting solutions for Java based application. Your requirement can be as simple as adding the "Print" menu in your standalone Java application or as complex as building a complete Web based interactive reporting system. Style Report has a solution for you!

- Flexible, word processor like layout
- Open data source including direct visual query for EJB, XML, JDBC/SQL, Corba
- Visual designer and query builder
- Built-in sorting, grouping, crosstab
- Built-in charting
- Output PDF, Postscript, CSV(Excel loadable), RTF(Word loadable) and HTML
- Integration with popular Java IDEs
- Support Java 1.1, 1.2 and 1.3

Style Report/Print

The printing engine that powers all Style Report products. With the advanced API, professional Java programmers have direct access to the full features of Style Report. Experienced programmers appreciate the flexibility provided by extensive use of design-patterns.

Style Report/Pro

Provides a powerful visual designer with multi-datasource query builder. It's not merely another report writer. Traditional banded report is just a component in the more flexible word processor like layout. Query building and report layout are separated in a model-view fashion Not only JDBC/SQL is data source, XML, EJB, JDBC/SQL, CORBA or even text files are data sources!

Download Free Lite Edition today!

www.inetsoftcorp.com

InetSoft

InetSoft Technology Corp • 559 Buckingham Dr Piscataway, NJ 08854 • Tel# (908)755-0200 • Fax# (908)756-1888

InetSoft, Style Report, Style Report/Enterprise Edition are trademarks or registered trademarks of InetSoft Technology Corp. Java, Swing and Sun are trademarks or registered trademarks of SUN Mircosystems, Inc. Adobe, Acrobat, PDF, Postscript are trademarks or registered trademarks of Adobe Systems Inc.

DevX and *Java*™ *Pro* present...

SPECIAL REPORT:

JUDGING JAVA™

A Comprehensive Analysis of the Past, Present, and Future of a Breakthrough Development Technology

Special Report Highlights

Part One: The State of the Java Market
Part Two: Your Next Steps in Java
Part Three: Java to the Max
Series Wrap Up (*Coming in August*)

Don't miss this special report on the state of Java's relevance right now and its real future in the Web-driven corporate development world. Presented by the Java experts at DevX, the know-how behind application development at www.devx.com, and *Java Pro*, the independent source for professional Java developers building enterprise-level business solutions.

Get your **FREE** Java Update newsletter subscription at: www.devx.com/free/newsletters

devX www.devx.com

Read ***Java Pro***— available on newsstands or by subscription. Call 1.800.848.5523 or 650.833.7100

JAVAPro www.java-pro.com

Special Report: Judging Java

Sponsored by: Rational® the e-development company™ take it to the n^th^ Sun microsystems

DevX is a trademark of DevX.com, Inc. Java is a trademark of Sun Microsystems, Inc. *Java Pro* is published by Fawcette Technical Publications, Inc. All other trademarks are property of their respective owners.

Scratching the Surface

Slick up your application's visual effects by using JSurface to manage panels with graphic borders and texture backgrounds

Sometimes all you need is a little decorating. This is especially true in multimedia applications and games, but many applications can benefit from slicker visual effects, so long as they are used in moderation. This month, we're going to make it possible to manage panels with graphic borders and texture backgrounds. We'll also demonstrate how you can make existing Swing components transparent in order to take advantage of textured backgrounds (see Figure 1).

To ensure flexibility, we'll use an interface to define the elements required by a JSurface component. The Surface interface allows us to access the background image and the elements for each side and corner. We'll develop a SurfaceImage class that implements this interface. By using an interface like this, we can allow different mechanisms to exist in the future, such as dynamically generated surfaces or surfaces from alternate data sources.

The Surface interface exposes the following methods:

```
public interface Surface
{
  public Insets getInsets();
  public Image
    getNorthWestImage();
  public Image
    getNorthEastImage();
  public Image
    getSouthWestImage();
  public Image
    getSouthEastImage();
  public Image getNorthImage();
  public Image getSouthImage();
  public Image getEastImage();
  public Image getWestImage();
  public Image getFillImage();
}
```

We expect to request any corner, side, or center image, as well as the Insets we'll use to draw with. The images, in the case of the fill or sides, will often be tiled to cover a larger area. We can provide a small 10x10 image, for example, and scale it to 100x100 by repeating sides horizontally or vertically and tiling the center image throughout the central area.

To define a suitable image, you must take this tiling into account and ensure that it will not produce undesirable edges. Fortunately, the Linux community has been developing themes for graphical environments for quite some time and provides a ready source of suitable images, though you are certainly free to create your own.

When you use JSurface, all you have to do is provide an image and a suitable Insets value, which defines the border area. In fact, because Swing already defines a suitable interface for supporting borders, we'll implement a SurfaceBorder class that you can use outside the JSurface component if you like, maximizing flexibility and reusability.

Figure 2 shows the classes we'll use in this project. Two utility classes implement static methods. The ImageUtil class implements loadImage(), toBufferedImage(), and tileImage() methods. The loadImage() method uses the ImageIcon implementation to create a blocking loader that returns a loaded Image. The toBufferedImage() method produces a BufferedImage from a normal Image ob-

by Claude **DUGUAY**

Claude Duguay is vice president of engineering at 360 Powered Corp. He has developed software for more than 20 years. He is passionate about Java, reads about it incessantly, and writes about it as often as possible. You can reach him at **claude@360.com**.

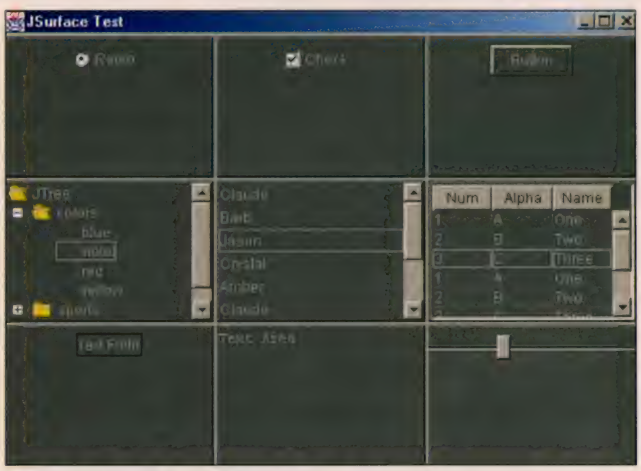


Figure 1 | On the Surface JSurface can be used to display transparent components on surface backgrounds.

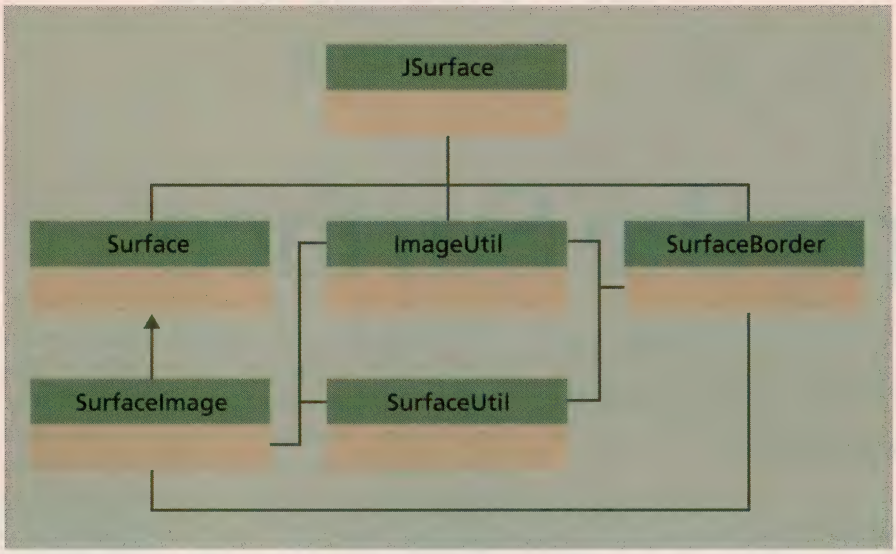


Figure 2 | Surface Classes Here are the classes used to implement JSurface. The ImageUtil and SurfaceUtil utility classes implement static methods.

ject. The tileImage() method tiles a specified image into a given Graphics context within a specified rectangular clipping area.

The SurfaceUtil class provides a set of methods to calculate rectangular areas from specified width and height values, and an Insets object. These calculations are required by both the SurfaceImage and SurfaceBorder classes, as well as by the JSurface component. None of these are very complicated but accessing them in a common class clarifies the rest of the code, making it considerably more maintainable. The choice of static methods over instance methods is somewhat arbitrary but follows the pattern established by the Java Math class for simple functions.

Having already seen the Surface interface and explained the code in ImageUtil

Listing 1 Surface Image

```
import java.awt.*;
import java.awt.image.*;

public class SurfaceImage implements Surface
{
  protected BufferedImage image;
  protected int width, height;
  protected Insets insets;

  public SurfaceImage(Image image, Insets insets)
  {
    width = image.getWidth(null);
    height = image.getHeight(null);
    this.image =
      ImageUtil.toBufferedImage(image);
    this.insets = insets;
  }

  protected Image getSubimage(Rectangle rect)
  {
    return image.getSubimage(
      rect.x, rect.y, rect.width, rect.height);
  }

  public Insets getInsets()
  {
    return insets;
  }

  public Image getNorthWestImage()
  {
    return getSubimage(SurfaceUtil.
      getNorthWestArea(width, height, insets));
  }

  public Image getNorthEastImage()
  {
    return getSubimage(SurfaceUtil.
      getNorthEastArea(width, height, insets));
  }

  public Image getSouthWestImage()
  {
    return getSubimage(SurfaceUtil.
      getSouthWestArea(width, height, insets));
  }

  public Image getSouthEastImage()
  {
  {
    return getSubimage(SurfaceUtil.
      getSouthEastArea(width, height, insets));
  }
  }

  public Image getNorthImage()
  {
  {
    return getSubimage(SurfaceUtil.
      getNorthArea(width, height, insets));
  }
  }

  public Image getSouthImage()
  {
    return getSubimage(SurfaceUtil.
      getSouthArea(width, height, insets));
  }

  public Image getEastImage()
  {
    return getSubimage(SurfaceUtil.
      getEastArea(width, height, insets));
  }

  public Image getWestImage()
  {
    return getSubimage(SurfaceUtil.
      getWestArea(width, height, insets));
  }

  public Image getFillImage()
  {
    return getSubimage(SurfaceUtil.
      getFillArea(width, height, insets));
  }
}
```

The SurfaceImage class implements the Surface interface. We use the getSubimage() method in BufferedImage to extract rectangular parts of the main image.

Listing 2 The Final Product

```
import java.awt.*;
import javax.swing.*;

public class JSurface extends JPanel
{
  protected Surface surface;

  public JSurface(String imageFile,
    Insets insets)
  {
    setOpaque(true);
    Image image = ImageUtil.loadImage(imageFile);
    surface = new SurfaceImage(image, insets);
    setBorder(new SurfaceBorder(
        surface, insets));
  }

  public void paintComponent(Graphics g)
  {
    int w = getSize().width;
    int h = getSize().height;
    Insets insets = getInsets();
    ImageUtil.tileImage(
      g, SurfaceUtil.getFillArea(w, h, insets),
      surface.getFillImage());
    g.setClip(0, 0, w, h);
  }
}
```

Much of the work in our JSurface class is handled by SurfaceImage and SurfaceBorder.

and SurfaceUtil, we'll concentrate on the SurfaceImage, SurfaceBorder, and JSurface classes in the rest of this column. I'll briefly mention the JSurfaceTest class and an undocumented TransparentRenderer included in the download, which you'll find at www.javapro.com. The TransparentRenderer implements the ListCellRenderer, TreeCellRenderer, and TableCellRenderer interfaces and is used to show how easy it is to create transparent Swing components.

The JSurfaceTest class is broken up into smaller methods that set various attributes in each of the Swing components that are demonstrated. This makes it easy for you to take a look at the methods and decide how you want to approach this in your own applications. Naturally, you could create transparent subclasses for each component, but I tried to keep things as simple as possible for demonstration purposes.

Listing 1 shows the code for the SurfaceImage class, which implements the Surface interface. The constructor stores the image width and height, along with the insets, and uses the toBufferedImage() method in ImageUtil to translate the image to a BufferedImage. This allows us to use the getSubimage() method in BufferedImage to extract rectangular parts of the main image. Because our SurfaceUtil methods return Rectangle objects, we implement a utility getSubimage() method to make this easier to work with.

The rest of the class implements the Surface interface, delegating most of the work to the SurfaceUtil methods and the getSubimage() method we implemented. The only additional method is getInsets(), which merely returns the instance variable we captured in the constructor.

> The techniques used in this month's implementation can be useful in various circumstances. If you need a more sophisticated solution, Swing makes it possible to plug in a custom look-and-feel

The code for SurfaceBorder, which implements the Swing Border interface, is available online. We save the Surface and Insets values in the constructor and return true for the isBorderOpaque() method. The insets value is returned by the getBorderInsets() method. Most of the work is done in the paintBorder() method and most of that is delegated to the ImageUtil and SurfaceUtil methods, drawing the corners and sides for each element, tiling the image as necessary for the sides.

Listing 2 shows the code for JSurface. As it turns out there isn't much work left for us to do because most of it is handled by SurfaceImage and SurfaceBorder. We implement a constructor that uses the provided image file name and Insets instance to create a suitable SurfaceImage. After using the loadImage() method from the ImageUtil class, we can create a SurfaceBorder and assign it to the JSurface component. We set the opaque value to true because we expect JSurface to fully cover anything behind it.

The paintComponent() method merely has to tile the fill image in the area within the border insets. We use the static getFillArea() method in the SurfaceUtil class to calculate the drawing rectangle and then reset the clipping rectangle after tiling the image. You'll notice that getting the Rectangle that represents the area inside border Insets is pretty common if you implement your own components. You may find the SurfaceUtil class and associated methods useful in alternate situations.

The techniques used in this month's implementation can be useful in various circumstances. If you need a more sophisticated solution, Swing makes it possible to plug in a custom look-and-feel. Some developers have already offered look-and-feel solutions that support textured backgrounds. Still, it's always good to understand how this might be done under more controlled circumstances and JSurface points the way to a few simple answers. Have fun with it. JP

GO ONLINE
www.javapro.com

Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109VC Download the code for this article.

JP0109VC_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

OO Databases Get Small Minded

The small, simple db4o object database is ideal for embedded applications, but is it too simple to be useful?

db4o is a very simple database engine whose entire API consists of a single factory object and a handful of interfaces (seven in all). The most recent version—version 1.1—includes only about 36 methods. You could just about figure out the whole thing in a single sitting.

The database engine is remarkably small and is therefore ideal for embedded applications. The JAR file for the executable takes less than 72K. It is a single-user system, hence it lacks support for locking and transactions. In addition, the company claims that the db4o engine is fast. (Benchmarks and benchmark results can be found on the company's Web site at www.db4o.com.)

db4o is a pure object database. No object-to-relational translation takes place, nor is there any need for a query language (given the size of the engine, there would be no space for either). Nevertheless, db4o manages to pack a great deal in a small space. It uses a novel variant of query-by-example—which I will explain shortly—to manage retrieval from the database. You can store multiple types of objects in the same database. The system even provides for callbacks, so an application can be notified when a particular object is stored, deleted, retrieved, and so on. The package is available for free from the company's Web site.

by Rick GREHAN

Rick Grehan is a technical writer with Compuware/NuMega. Rick was a senior editor at *BYTE* magazine (where he wrote the JavaTalk column) and at *Computer Design* magazine. He can be reached at rgrehan@top.monad.net.

Query By Example

Query-by-example—often referred to simply as QBE—is a technique for querying a database that does not use a query language (such as, for example, SQL) to specify the query. Instead, QBE uses a template to define the query. The form that this template takes can vary depending on the database system.

Some QBE systems (Borland's Paradox, for example) provide what amounts to a tabular view that mimics the row-and-column structure of the database to be queried. Each cell in this template view corresponds to fields in the database. When you enter a query expression into a given cell, that expression applies to the associated field in the database. Usually, these expressions are simple conditional equations that specify the range of field values that constitute a successful query match.

In db4o, query by example is accomplished via a template object. The process is extremely simple, and involves a single method. Objects that satisfy the query are returned in a navigable container object, as I'll explain shortly.

Inside db4o

The single class in the db4o API is Db4o. It serves as a factory for the workhorse objects of the system. For example the Db4o.configure() method returns a Configuration context (an object that implements the configuration interface). Via this Configuration context, an application can modify the behavior of the database engine.

The Db4o object's openFile() method returns an ObjectContainer object. The openFile() method effectively opens the database, and the returned ObjectContainer object provides the methods whereby the application manipulates the database.

For example, suppose I had created a database file "mydata.db," which is stored in the local directory. I would open that database with a call that looks like:

```
ObjectContainer myDB =
  Db4o.openFile("mydata.db");
```

The myDB object returned provides all the methods needed to access the database:

- get()—This method reads one or more objects from the database. This is also the method for querying the database.
- set()—This method stores an object in the database.
- delete()—This method removes an object from the database.

The ObjectContainer object also provides a number of utility methods. Three of the more important ones are:

PointBase

PointBase Delivers:

Faster time to market and low cost of deployment...through the use of 100% Pure Java™ and adherence to mainstream industry standards including SQL, JDBC, TCP/IP and HTTP. PointBase enables deployment on any platform with no dependency on the OS version or hardware.

Ultra-small resource requirement with dynamic footprint architecture... PointBase's dynamic footprint architecture ensures that features are loaded into memory only as required. This modular deployment of functionality optimizes the use of memory and processor resources. The PointBase JAR™ footprint can be as small as 45K.

Near zero administration...designed for simple one-time configuration to enable transparent integration with applications and large-scale deployment, PointBase uses statistics collected at run-time in its query optimizer to adaptively tune the system for maximum application performance, eliminating the need for manual intervention.

Universal synchronization with enterprise systems...PointBase supports transparent, bi-directional synchronization with major enterprise databases including Oracle, IBM DB2, Sybase and Microsoft SQL Server.

Available Now...

PointBase 4.0 and PointBase Micro

www.jp1.pointbase.com

1-877-238-8798

PointBase

• commit()—This method writes all cached objects to the database.
• isStored()—The isStored() method takes as an argument an object reference and returns a boolean indicating whether the object is stored in the database.
• storedClasses()—This method returns a string array of the classnames of all the stored classes in the database.

Querying and Template Objects

As mentioned previously, the get() method within the ObjectContainer interface is the query method for the database. The get() method takes a single java.lang.Object argument, which is referred to as the template object. This object controls how the get() method performs both query and retrieval.

A template object is nothing more than an "ordinary" object whose fields have been preloaded with values that define the retrieval criteria. Necessarily, the template object must be an object of the same class as those objects you wish to fetch from the database. The db4o engine will retrieve from the database all objects whose field contents match the contents of the fields in the template object. Fields within an object that are references to other objects and that hold a NULL are ignored for the purpose of comparison. Also, primitive fields in the template object whose contents are zero or false are ignored.

> The db4o database engine uses a novel variant of query-by-example

Here's where db4o's simplicity spawns some significant limitations. Notice that db4o's form of QBE permits only equality comparisons. The system will retrieve only objects whose fields match those of the template. Suppose, for example, I had an Employee class that included a floating-point salary field. I cannot, in a single db4o query, retrieve those employees whose salaries are between $50,000 and $60,000. (In fact, I can't even do it in a fixed number of queries. I would have to retrieve all Employee objects and filter them after retrieval.)

Even worse is the fact that db4o must use the zero or false value in primitive fields as an equivalent to null. So if, for example, my database application stored the number of dependents each employee had, I could not create a db4o query that returned all employees who had zero dependents.

Retrieved Objects and Activation

Issuing a get() method with a particular template object returns an object that implements the ObjectSet interface. As its name suggests, an ObjectSet is a collection of

Listing 1 Building the Database

```
public class Employee
{
  public String name;   // Employee name
  public int socsecnum; // Employee ssn
  public float salary;  // Employee salary

  public Employee() { }

  public String toString()
  {
    return("Name: "+name+" ID: "+
      Long.toString(socsecnum)+
      " Salary: "+Float.toString(salary));
  }
}

public class Manager extends Employee
{
  public String projectname;
  public int numemployees;

  public Manager() { }

  public String toString()
  {
    return(super.toString()+" Proj: " +
      projectname);
  }
}

import com.db4o.*;

// Create a database
public class load_db
{
  public static void main(String[] args)
  {
    // Objects to store
    Employee tEmp;
    Manager tMgr;

    // Create the database
    ObjectContainer mydb =
      Db4o.openFile("employees.db");

    // Build some employees
    // and store them
    tEmp=new Employee();
    tEmp.name="Tom";
    tEmp.socsecnum=111111111;
    tEmp.salary=41000.00f;
    mydb.set(tEmp);
    tEmp=new Employee();
    tEmp.name="Dick";
    tEmp.socsecnum=222222222;
    tEmp.salary=43000.00f;
    mydb.set(tEmp);
    tEmp=new Employee();
    tEmp.name="Harry";
    tEmp.socsecnum=333333333;
    tEmp.salary=43000.00f;
    mydb.set(tEmp);

    // Build a manager and
    // store her
    tMgr=new Manager();
    tMgr.name="Sarah";
    tMgr.socsecnum=444444444;
    tMgr.salary=43000.00f;
    tMgr.projectname="Development";
    tMgr.numemployees=3;
    mydb.set(tMgr);

    // Close the database
    mydb.close();
  }
}
```

This source code builds the database used in the query example shown in Listing 2.

IBM is a registered trademark and developerWorks is a trademark of International Business Machines Corporation.
Java is a trademark of Sun Microsystems, Inc. Linux is a registered trademark of Linus Torvalds. © 20

THE CODERNAUTS FIND A BETTER WAY TO KEEP UP.

developerWorks™ *FOR* YOUR BRAIN

TOOLS • CODE • EDUCATION • EMERGING TECHNOLOGIES • LINUX® • XML • JAVA™

ibm.com/developerWorks

objects. The db4o ObjectSet interface includes methods that let you iterate through the members of the set, as well as methods that return the size of the ObjectSet (or a boolean value that indicates whether the set is empty). If you issue a get() method whose template object is null, db4o will return an ObjectSet whose contents are the entire contents of the database.

> Possibly db4o's greatest limitation is the fact that its query supports only "equals" comparisons

An object retrieved from a db4o database is instantiated, but not necessarily activated. That is, memory is allocated for the object when it is retrieved, but until the object is activated, the object's fields are set to default values. And if a particular field is a reference to another object, that field is in an indeterminate state.

The db4o database engine, by default, activates objects five levels deep. So, if object A references object B, which references object C, and so on to object E, reading object A from the database will instantiate and activate objects A through E. However, if object E references object F, object F will be instantiated but not activated.

You activate an object by calling the ObjectContainer method activate() with the object to be activated as the argument. This performs a get() on all the object's member objects. Note, however, that although the member objects are instantiated, they are themselves in a deactivated state.

Now, three paragraphs back I said that referenced objects would "not necessarily" be activated. I said that because db4o lets you set a global activation depth. This is one of the methods in the Configuration context object mentioned previously.

ObjectClasses and ObjectFields

As I've already shown, you can perform global configuration of database characteristics using the Configuration context. The db4o system also lets you configure the behavior of specific classes within the database. You do this with the help of an ObjectClass object, which you can obtain through the configuration context class as follows:

```
Configuration cfig =
  Db4o.configure();
ObjectClass objc =
  config.objectClass(
  "<classname>");
```

where <classname> is the class's fully qualified name.

Once an ObjectClass object is obtained, you can, for example, set the minimum and maximum activation depth for the target object's class. You can also specify a method within the class to be called when an object of the class is activated.

An ObjectField object—which can be acquired via a factory method within ObjectClass—lets you configure the behavior of individual fields. Specifically, you can toggle whether or not a particular field participates in the query. This at least minimizes the amount of work an application must do to set up the template object.

Query Example

To get a feel for working with db4o, I've created a couple of simple programs. The first, shown in Listing 1, creates a database that holds both Employee and Manager objects (Manager is a subclass of Employee). The program simply creates three Employee objects and a single Manager object, and loads them all in the database.

Listing 2 is a simple query program that accepts three command-line arguments. These command-line arguments are loaded into the query template object. The application opens the database created by Listing 1, issues the query, and displays the list of returned objects.

As an example, the following shows how to use dbQuery to query for a particular employee based on social security number:

```
C:\db4o>java dbQuery null
  444444444 0.0f
Name: null ID: 444444444
  Salary: 0.0
# in result set: 1
Name: Sarah ID: 444444444
  Salary: 43000.0
  Proj:Development
```

Listing 2 What's Your Query?

```
import com.db4o.*;
import java.lang.*;

// Query the db4o database
public class dbQuery
{
  public static void main(String[] args)
  {
    // Template object
    Employee tEmp = new Employee();

    // Load the template object
    if(args[0].equalsIgnoreCase("null"))
      tEmp.name=(String)null;
    else
      tEmp.name=args[0];
    tEmp.socsecnum =
      Integer.parseInt(args[1]);
    tEmp.salary =
      Float.parseFloat(args[2]);

    // Open the database and do the query
    ObjectContainer mydb =
      Db4o.openFile("employees.db");
    ObjectSet myos = mydb.get(tEmp);

    // Iterate through the query set
    // and output
    System.out.println(
      "# in result set: " + myos.size());
    while(myos.hasNext())
      { Object myobj = myos.next();
      System.out.println(
      myobj.toString());
    }
    mydb.close();
  }
}
```

This query program accepts three command-line arguments that are loaded into the template object. Notice that using "null" for the Employee.name field will set a null reference in that field.

A similar query returns all employees whose salary equals $43,000:

```
C:\db4o>java dbQuery null
  0 43000.0f
Name: null ID: 0 Salary: 43000.0
# in result set: 3
Name: Harry ID: 333333333
  Salary: 43000.0
Name: Dick ID: 222222222
  Salary: 43000.0
Name: Sarah ID: 444444444
  Salary: 43000.0
  Proj: Development
```

Notice in both of the preceding queries, the numeric fields that were set to zero matched all cases in the database. You can see this in the extreme if you set all the query fields to null and zero:

```
C:\db4o>java dbQuery null 0 0.0f
Name: null ID: 0 Salary: 0.0
# in query set 4
Name: Harry ID: 333333333
  Salary: 43000.0
Name: Dick ID: 222222222
  Salary: 43000.0
Name: Tom ID: 111111111
  Salary: 41000.0
Name: Sarah ID: 444444444
  Salary: 43000.0
  Proj: Development
```

This returns all items in the database.

I was initially interested in db4o because of its applicability to handheld devices. The db4o database engine does come in a form that will execute on Symbian/EPOC devices. Though db4o has a small enough footprint to fit on a Palm handheld device, it cannot run on the PalmOS—at least not yet. As of this writing, no VM for the PalmOS supports reflection—a capability that db4o requires to work its magic. (The Kada virtual machine may solve this problem in the near future.)

Possibly db4o's greatest limitation is the fact that its query supports only "equals" comparisons; you can only query/retrieve objects whose fields equal the values in the associated fields of the template object. The engineers at db4o are investigating additions to the API that supports more complex queries. Work in this direction can be found at the Web site www.odbms.org/soda, which presents the SODA (simple object database access) query API. JP

GO ONLINE
www.javapro.com

Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109JG Download the code for this article.

JP0109JG_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

WEBBuilder

NEW ORLEANS
NOVEMBER 28-30

Bring Your Entire Internet Team to the

Next Generation Internet Event

Web Builder is the ONLY event that joins ASP.NET and Java programmers with Web and interaction designers. Web Builder equips your entire team to build the Web sites and services of 2002.

Three Full Conferences in one MASSIVE event!

1 Web Design World 2001

2 JAVA TECH SUMMIT

3 ASP.net TECH SUMMIT

http://conferences.devx.com/WebBuilder 800-848-5523

Web Builder, Java Tech Summit, ASP.NET Tech Summit, Web Design World, *Java Pro Magazine*, *Visual Studio Magazine* are trademarks of Fawcette Technical Publications, Inc. Java is a trademark of Sun Microsystems. All other trademarks are property of their respective owners.

Continued from page 21.

and observe real-time operating system requirements. It tries to map Java threads to native threads whenever possible and can load classes out of ROM. This is not a virtual machine you would run on a phone or Palm device. It is better suited to game consoles and set-top boxes than handheld computers.

CLDC is much simpler than CDC. In fact, CDC is a superset of CLDC, containing all of its class libraries. This was a smart move and likely the case because the JSR-36 and JSR-46 specification leads are one and the same person and many of the same companies participated in both working groups. This means that all CLDC code will run in a CDC environment. However, as soon as you start using the classes in different profiles, all bets are off.

CLDC is a lot more restrictive than CDC. A CDC JVM, such as the KVM, does not support floating point numbers. This was required because most of the targeted devices do not have floating point hardware. CDC JVMs also do not support finalization and restrict subclasses of Error to a minimal set. The lack of floating point numbers is probably the greatest limitation. Additional limitations include no JNI support, no reflection, no weak references, no thread groups, no daemon threads, and no user-defined class loaders. Furthermore, compiled classes must be run through a preverifier before they can be used by the virtual machine. So you can't just download a J2SE class file from the network and expect it to work.

The set of libraries contained in a CLDC environment is also restricted. It includes only four packages: java.io, java.lang, java.util, and javax.microedition.io. Any code you write using the javax.microedition.io package will work in a CDC environment as well, but not in J2SE. The java.lang package is rather bare, stripped down to the basic boxing types (for example, Integer, Long, and so on), String, StringBuffer, Thread, and a few others. java.util is even more bare. It contains only the Enumeration interface, the Stack, Vector, and Hashtable containers, the Date, Calendar, and TimeZone time-handling classes, and Random. Forget about using the Collections Framework. If Vector, Stack, or Hashtable can't do what you want, write your own classes to do the work. java.io is limited to a few stream and reader/writer classes, while javax.microedition.io provides only the Connector class and some supporting interfaces. Connector contains a set of methods that provides a device-independent way of accessing data streams, be they on the network or local. You will have to rewrite all socket or file-based I/O to use the Connector class for it to work with CLDC.

> Figuring out how to port a J2SE project to J2ME is not all that difficult—the tough part is deciding on software development and maintenance strategies for cross-edition and cross-profile Java projects

Profiles

Profiles pick up where configurations leave off. They specify additional functionality for subsets of devices in a configuration. For example, a game console and an advanced PDA may fall under CDC, but neither device needs all of the functionality required by the other, so different profiles would be defined for game machines and advanced PDAs.

At the moment, two profiles have been defined for CDC and three more are in the works. The Foundation Profile fills in the rest of the differences between CDC and J2SE, adding TCP socket networking classes, internationalization support, zip compression, and the remaining classes in the java.lang, java.io, and java.util packages. The RMI Profile adds Remote Method Invocation classes built on top of the java.net package. The specification has been released, but a reference implementation is not yet available. The Personal Profile and the Personal Basis Profile implement the next generation of what used to be PersonalJava. The Personal Basis Profile will add minimal graphical interface support based on AWT, and the Personal Profile will expand on it to provide full compatibility with previous PersonalJava implementations. The Game Profile will add APIs specifically for high-end game consoles and desktops, so don't count on it helping you write games for mobile phones. No specifications have been released yet for these last three profiles.

A CDC environment with the Foundation and RMI Profiles looks a lot like a GUIless J2SE implementation and doesn't provide many porting challenges for nongraphical code. For example, many of the libraries from the Apache Jakarta Project will compile for CDC and the Foundation Profile without modification.

CLDC profiles are a different story. The Mobile Information Device Profile (MIDP) does not add any more classes to the core packages beyond java.lang.IllegalStateException, java.util.Timer, and java.util.TimerTask. However, it does add the javax.microedition.lcdui package that looks like it will form the basis for all user interface programming in CLDC. MIDP also adds a record management package in the form of javax.microedition.rms as well as a base Midlet application class in the javax.microedition.midlet class. Right now you want to throw away your KVM Spotlets and write CLDC Midlets instead. The Spotlet APIs that shipped with the KVM demonstration implementation for the Palm won't be supported. Neither the PDA Profile nor the MIDP Next Generation working groups have released specifications, but it is clear that both will be built directly on top of MIDP. Your MIDP code will work with both profiles. The Wireless Telephony Communication APIs (WTCA) are not a profile in and of themselves, but are rather a horizontal set of APIs for telephony applications that can be incorporated into both CDC and CLDC profiles.

The Porting Challenge

Figuring out how to port a J2SE project to J2ME is not all that difficult. Just compile it with the target configuration and profile reference and see what breaks. The tough part is deciding on both software development and maintenance strategies for cross-edition and cross-profile Java projects. This is made more difficult by Java's lack of a preprocessing mechanism. I'm not going to recommend a particular preprocessing product, but I will say that you need a

Listing 1 Conditional Networking

```
#if defined(ENABLE_J2ME_CLDC)
  import javax.microedition.io.*;
#else
  import java.net.*;
#endif
  import java.io.*;

public class OpenURL {

  #if defined(ENABLE_J2ME_CLDC)
    public static final InputStream openURL(
      String url)
  throws IOException {
    Connection connection = null;

    return Connector.openInputStream(url);
  }
  #else
    public static final InputStream openURL(
      String url)
  throws IOException {
    URL connection;

    connection = new URL(url);

    return connection.openStream();
  }
  #endif

}
```

The use of configuration-specific APIs can be encapsulated in higher-level functions or classes. In this example, the C preprocessor is used to allow CLDC and CDC/J2SE implementations of a function to coreside in the same source file.

preprocessor for conditional compilation. It is simply not practical to use elaborate factory classes, introspection, and other runtime mechanisms to produce all platform-specific behavior. If all you have to change is one or two lines of code to get a class to work in another configuration, that's what #ifdef is good for. You will have to decide on how best to isolate configuration-specific behavior. Some options include conditional compilation, subclassing, factory classes, and new reimplementations of classes. Source code organization will also become a problem, because you may need to reimplement a class for a particular configuration in a separate source file. These are configuration management problems that can be handled by a little shell-scripting and clever use of make or Ant.

Making code work with CDC and the Foundation Profile is not worth talking about too much because it's not very involved. Most J2SE code will work without modification. The hangups will be with the GUI code. Until the Personal Profile and Personal Basis Profile are released, it will be difficult to say what will fly. It is probably safe to say that a full Swing implementation will not be available in at least the Personal Basis Profile.

Whether you write for CDC or CLDC, it is clear that you will have to separate your application logic from your application display as much as possible. Revisit the Model-View-Controller pattern to practice the technique. It's easy to get sloppy and mix a lot of logic inside of user interface code for the sake of getting the job done quickly. You can't do that if you want your code to work with both J2SE and J2ME unless you like duplicating a lot of code. There's no doubt that companies will want to write programs, like a personal calendaring application, that will work across CLDC, CDC, and J2SE. Application entry and exit points will sometimes be different, such as in the case of MIDP, where an application must extend the Midlet class. Count on writing at least two, if not more, user interfaces for each application.

Outside of user interfaces, networking and I/O is the next major area of concern, primarily for CLDC. Networking and I/O for J2SE and CDC are pretty much the same. For example, the code from the Pro Shop column "Build Your Own Web Server" (*Java Pro*, July 2001) will work without modification when compiled and run with CDC and the Foundation Profile. CLDC doesn't have sockets, a notion of files, or a rich set of streamhandling classes. You will have to implement most networking and I/O code at least twice—once for J2SE and once for CLDC. Depending on the program, either the J2SE or CLDC code could be used for CDC. Given that most of this code works with streams, you should try to write the core logic once, so that it operates only on InputStream, OutputStream, Reader, or Writer. The connection establishment code that returns a stream reference, such as opening a file or connecting to a network resource, will require separate implementations. But if you encapsulate it properly, code that depends on it will not have to change.

A simple example highlights some of the cross-edition programming issues (see Listing 1). In the example, I've chosen to use the C preprocessor, which should be familiar to most programmers, to allow two implementations of a function to exist in the same source file. The first implementation uses the javax.microedition.io package to open an InputStream from a URL. This function will work in J2ME, CLDC, or CDC, but not J2SE. The second function implementation will work in J2ME CDC and J2SE. By encapsulating the URL opening code into a conditionally compiled function, code that depends on the functionality does not have to be changed.

Ideally, the splintering of Java into multiple editions, configurations, and profiles will motivate Sun to add conditional compilation support to the language environment. Assertions have already been added to the language in JDK 1.4, incorporating a conditional code execution feature so that assertions can be turned on or off at run time. If such a feature were generalized to allow arbitrary flagged bytecode to be conditionally executed, that would provide the ability to ship one class file for all Java environments. Until such an initiative appears in the Java Community Process, Java developers will have to cook up their own strategies for maintaining portable Java code. JP

Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109DS Download the code for this article.

JP0109DS_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

Generics in Java

The better-late-than-never generics extension to Java promises to provide language-intrinsic support for type-safe generic programming

by Daniel F. **SAVARESE**

C++ was born sometime around 1983 and did not become an ISO standard until 1998. Those 15 years saw many changes made to the language and its standard library. A lot of thought went into designing the language and much of its complexity is rooted in satisfying the practical needs of software development projects. Just as C++ did not spring fully formed from the head of Bjarne Stroustrup, Java did not leave James Gosling's care as an adult.

Java has already experienced several waves of language changes and API additions and revisions. I expect many more over the next 10 years. Unfortunately, many of the changes in Java have been necessitated because of the haste in getting the language into developers' hands in 1995, shortsightedness in the design of many of the APIs, sheer carelessness by Sun in its custodianship of the language, and Java's caretakers not paying attention to the lessons learned during the standardization of C++.

The fifth iteration of the Java Developer's Kit, JDK 1.4, is on the horizon, and with it comes many more changes. If there is a sin that Sun keeps on repeating, it is the continued bloating of the core APIs and the constant correction of avoidable mistakes. (I'd rather they not make the mistakes in the first place.) As an example of bloat, I offer the new java.util.regex package, stemming from JSR-51. Not only do a plethora of third-party offerings provide the functionality, but the provided API has an ill-conceived design that will likely find itself deprecated and corrected in JDK 1.5.

Sun continues to marginalize third parties by introducing non-essential functionality into the core APIs, violating the philosophy espoused by James Gosling and Guy Steele of keeping Java as small as possible and letting others extend it with their own libraries. As an example of correcting past mistakes, I offer the java.nio package, also stemming from JSR-51, which provides fast I/O classes. Java's I/O capabilities have been notorious for their poor performance and limited ability to perform essential tasks such as file locking. Let's hope this attempt to fix the situation is successful.

In a keynote address at OOPSLA '98 (the ACM Conference on Object-Oriented Programming, Systems, Languages, and Applications), Guy Steele said "... the Java programming language needs to grow yet some more—but, I hope, not a lot more. At least, I think only a few more rules are needed—the rest can be done with libraries, most of them built by users and not by Sun." (You can download a copy of his interesting address from the Generic Java Web site at http://www.cs.bell-labs.com/who/wadler/pizza/gj/Documents/steele-oopsla98.pdf.) Even as Sun has ignored this guidance by adding monstrous numbers of libraries to the Java platform, it has heeded it reasonably well by limiting the number of additions to the language proper. We are now close to seeing a very important addition made to the language that all Java developers should pay attention to: language-intrinsic support for type-safe generic programming.

> Even as Sun has ignored guidance by adding monstrous numbers of libraries to the Java platform, it has heeded it reasonably well by limiting the number of additions to the language proper

Java may be a strongly typed language, but calling it type-safe is a bit of a stretch. I have been complaining about the constant need to perform dynamic casts from Object since 1995. Using dynamic casting as the basis of generic programming produces fragile code that may fail at any moment with a ClassCastException. Dynamic casting is also inefficient, incurring runtime overhead. C++ provides compile-time type-checking of generic programs using the template construct. This static type-checking ensures that your generic algorithms will behave in a predictable manner. If you accidentally add a String to a list of Integers, the compiler will catch the error rather than delaying it until runtime. In Java, the program would compile just fine, but it might unexpectedly fail at runtime.

Many programmers have bemoaned Java's lack of facilities for type-safe generic programming, but a research group at Bell Labs actually did something about it, developing the Pizza compiler (http://www.cs.bell-labs.com/who/wadler/pizza) in the JDK 1.0.2 days. The Pizza compiler added parametric polymorphism to the Java language, akin to C++ templates, as well as other features such as first-class functions that could be referenced by variables. That work was subsequently refined, becoming the Generic Java project (http://www.cs.bell-labs.com/who/wadler/pizza/gj), which finally resulted in JSR-14 (http://jcp.org/aboutJava/communityprocess/review/jsr014/index.html). The Java Community Process is producing some questionable results, but in this case, it is to be lauded for enabling the addition of an essen-

tial language feature pioneered by a group entirely outside of Sun.

An early access release of the generics extension to Java is available from the Java Developer Connection at http://developer.java.sun.com/developer/earlyAccess/adding_generics. The JSR-14 public draft indicates the extension was intended to be added in JDK 1.4, but that it will have to wait until JDK 1.5 because of some changes required to the JVM to better support reflection. Regardless, you can start using the features today because the compiler in the early access release generates bytecode that is compatible with J2SE 1.3. Strangely enough, I couldn't use J2SE 1.3 to run the compiler because it appeared to have a dependency on the CharSequence class, which is only present in the 1.4 beta release. I had no problems when I used JDK 1.4 beta to run the JSR-14 early access compiler and either JDK 1.3 or JDK 1.4 beta to execute the resulting class files.

You can get a feel for what the generics extension buys you by comparing Listings 1 and 2. Listing 1 shows how generic programming in Java works today. It implements both a min and a max function, which return the smaller and the larger of their arguments respectively. On the surface the functions seem relatively safe. They each accept arguments and return values that implement the Comparable interface, but therein lie some problems.

> The Java Community Process is to be lauded for enabling the addition of an essential language feature pioneered by a group entirely outside of Sun

The first problem is that objects implementing the Comparable interface must be compatible if they are to be compared, but their compatability is only evidenced at runtime. The main() method shows how comparing incompatible types—in this case a String and an Integer—can cause a program to fail at runtime by throwing a ClassCastException. The second problem is that the return value must be dynamically cast to another type in order to be useful, introducing another weak link.

Listing 2 shows how the same functions would be written using the generics extension. The functions are now parameterized, using a syntax similar to C++ templates, allowing type-checking to occur at compile

Listing 1 Generic Methods Today

```
package example;

public class MaxMin {
  public static
  Comparable max(Comparable obj1,
    Comparable obj2) {
    if(obj1.compareTo(obj2) >= 0)
      return obj1;
    return obj2;
  }

  public static
  Comparable min(Comparable obj1,
    Comparable obj2) {
    if(obj1.compareTo(obj2) <= 0)
      return obj1;
    return obj2;
  }

  // This program will throw a ClassCastException
  public static final void main(String[] args) {
    String max;
    Integer min;

    max = (String)max("Bilbo", new Integer(111));
    min = (Integer)min(new Integer(33), "Frodo");
    System.out.println(max + " " + min);
  }
}
```

Today, generic programming in Java is not type-safe and depends a lot on the use of interfaces and dynamic casting. This program will compile without errors but fail with a ClassCastException at runtime.

Listing 2 Generic Methods Tomorrow

```
package example;

public class MinMax {
  public static <Type extends Comparable<Type>>
    Type max(Type obj1, Type obj2)
  {
    if(obj1.compareTo(obj2) >= 0)
      return obj1;
    return obj2;
  }

  public static <Type extends Comparable<Type>>
    Type min(Type obj1, Type obj2)
  {
    if(obj1.compareTo(obj2) <= 0)
      return obj1;
    return obj2;
  }

  public static final void main(String[] args) {
    String max;
    Integer min;

    max = max("Frodo", "Bilbo");
    min = min(new Integer(33), new Integer(111));
    System.out.println(max + " " + min);
  }
}
```

Generic programming extensions allow you to write type-safe methods. The min() and max() methods rely on a parameterized version of the Comparable interface to define the types of their arguments and return values. The main() method from Listing 1 would produce a compiler error if used with these methods.

Listing 3 Generic Classes

```
package example;

public interface Addable<Type> {
  public Type add(Type t);
}

package example;

public class MyNumber implements Addable<MyNumber> {
  Number number;

  public MyNumber(Number n) {
    number = n;
  }

  public double value() {
    return number.doubleValue();
  }

  public MyNumber add(MyNumber n) {
    return new MyNumber(new Double(value() +
      n.value()));
  }
}

package example;

public class MyString implements Addable<MyString> {
  String string;

  public MyString(String s) {
    string = s;
  }

  public String value() {
    return string;
  }

  public MyString add(MyString s) {
    return new MyString(value() + s.value());
  }
}

package example;

import java.util.*;

public class Total<Type extends Addable<Type>>
  extends ArrayList<Type>
{

  public Type total() {
    Type sum = null;
    Iterator<Type> it;

    it = iterator();

    if(it.hasNext())
      sum = it.next();

    while(it.hasNext())
      sum = sum.add(it.next());

    return sum;
  }

  public static final void main(String[] args) {
    Total<MyNumber> numbers =
      new Total<MyNumber>();
    Total<MyString> strings =
      new Total<MyString>();

    numbers.add(new MyNumber(
      new Double(0.18485)));
    numbers.add(new MyNumber(new Integer(2)));
    numbers.add(new MyNumber(new Integer(8)));

    System.out.println(numbers.total().value());

    strings.add(new MyString("Hello"));
    strings.add(new MyString("World"));
    strings.add(new MyString("!"));

    System.out.println(strings.total().value());
  }
}
```

Classes, as well as methods, may be parameterized, but the lack of operator overloading and support for primitive types makes some programming tasks awkward. Here, the Total class tries to generically sum the elements of a list.

time. The calls to min() and max() from Listing 1 would now fail at compile time rather than run time.

We can learn a few other things from Listing 2. First, notice the syntax for defining a parameterized method. The parameterized types used by the method must first be defined in between angle brackets before being used as part of the method declaration and implementation. Second, a parameterized type may be specified as implementing an interface or extending a class. Be careful when trying this out. The examples in the public draft specification indicate that the keyword to use for denoting interface implementation is *implements*, but in fact, only *extends* is recognized. The third thing to note is that the Comparable interface itself is parameterized. As part of the generics extension, the entire Collections Framework is made to work with parametric types.

Listing 3 shows how to write parametric classes and interfaces and also reveals some of the limitations of the generics extension. The Addable interface is used because Java does not support operator overloading, yet another feature of C++ that we will probably see adopted by Java in the future. MyNumber wraps instances of Number so that they may be generically added and MyString implements string concatenation as part of the Addable interface. The Total class implements a list that can add up its contents to produce a total. We've already seen that the lack of operator overloading is an obstacle to facilitating this process. Another obstacle is the dichotomy between user-defined and primitive types. C++ templates are implemented using what is sometimes called *heterogeneous translation*. This means that each template instance produces a separate class. Java generics use *homogeneous translation*, where each template instance uses the same class, called an *erasure* because it's what remains after you remove all the parameters from a type definition. This is necessary for many reasons, the foremost being the maintenance of both backward and forward compatibility.

Java generics are primarily syntactic sugar that allow the compiler to do some static type-checking. Underneath, the compiler generates the same kind of code you would write yourself, complete with ugly runtime casts. The difference is that the compiler will ensure that no casting mistakes are made. The downside is that you can't support primi-

tive types without boxing them with Integer, Float, and so on. Even though the compiler could do this boxing for you automatically, it would be extremely misleading to have it do so because it makes it seem as though you're using an efficient mechanism when what you're really using is an abhorrently inefficient mechanism. The JSR-14 public draft contains a detailed discussion of why primitive types are not supported. The designers covered every angle I have considered in the past, but unfortunately were not able to come up with a solution that would fit all of the design constraints. Once a software system, in this case the Java 2 Platform, is in wide use, it's impractical to do things that break it. My main interest in adding generics to Java has always been to be able to write code that worked with both user-defined objects and primitive types—and efficiently in both cases. I'm not the only one who feels this way, so there will be a lot of disappointed Java programmers. However, the design and implementation of the generics extension does not preclude the future addition of primitive type support.

I could say a lot more about the generics extension, but you can get most of the information you need from reading the JSR-14 public draft specification. The generics extension is different from other proposed additions in that it is highly influenced by the dynamic nature of the Java runtime environment.

> **Java generics are primarily syntactic sugar that allow the compiler to do some static type-checking**

Deciding on the proper way to implement generics in Java is not as simple as looking at what was done in C++, and the JSR-14 working group has done a commendable job of analyzing the problem. However, many other additions are no-brainers. They will inevitably find their way into the language as Java continues to take over the role of C++. These include enumerations, true conditional compilation, and operator overloading, just to name a few. Many hard lessons were learned during the 15 years of the C++ standardization process. Unfortunately, most of them are being relearned during the sometimes haphazard Java non-standardization process. **JP**

Daniel F. Savarese is an independent software developer and technology advisor. He has been the founder of ORO, Inc., a senior scientist at Caltech's Center for Advanced Computing Research, and vice president of software development at WebOS. Daniel is the original author of the Jakarta-ORO text processing packages and the NetComponents network protocol library. He is also a coauthor of *How to Build a Beowulf* and is working on several computer science book projects. He can be reached at **dfs@savarese.org**.

Use the following DevX Locator+ codes at www.javapro.com to go directly to these related resources.

JP0109 Download all of the code for this issue of *Java Pro*.

JP0109PS Download the code for this article.

JP0109PS_T Read this feature online. DevX Premier Club membership is required.

Not a member yet? Go to www.DevX.com to subscribe to the Premier Club.

Advertiser Index

Actuate Corporation 46
800-884-8665
www.actuate.com/info/jpad.asp

AireWeb 51
866-AIREWEB
www.aireweb.com

BEA Systems 4
800-817-4BEA
www.bea.com

Borland 27
800-331-0877
www.borland.com

Cape Clear Software 19
866-CAPE-226
www.capeclear.com

Colorado Software Summit 35
800-481-3389
www.softwaresummit.com

Compuware Corporation 53, 61
www.compuware.com

DevX.com 68
888-963-3389
www.devx.com

DevX Knowledge Xchange 67
888-963-3389
www.devx.com/knowledgexchange

DevX MarketPlace 95
888-963-3389
marketplace.devx.com

Dynalivery 55
800-662-7771
www.dynalivery.com

Embarcadero Technologies 58, 59
www.embarcadero.com

Empirix 45
www.empirix.com

ESRI 31
888-289-5084
www.esri.com/arcims

Fiorano 29
www.fiorano.com

IBM Corporation 71
www.ibm.com/websphere/mankind

IBM Corporation 73
ibm.com/db2/faster

IBM Corporation 85
ibm.com/developerworks

Inetsoft 77
www.inetsoftcorp.com

Infragistics, Inc. 2, 3
800-231-8588
www.infragistics.com

InstallShield Software Corp. 41
www.installshield.com

IONA Technologies 8, 9
www.iona.com

Java Pro Magazine Subscriptions 33
800-848-5523
www.javapro.com

Java Tech Summit 25
800-848-5523
conferences.devx.com/webbuilder

Judging Java 78
888-963-3389
www.devx.com

LOOX Software 17
800-684-5669
www.loox.com

NeuNet Solutions, Inc. 22
916-424-5870
www.neunetsolutions.com

Object Design 10
www.objectdesign.com

OpenMarket 36, 37
800-OPEN-MKT
www.openmarket.com

Parasoft Corporation 38
888-305-0041
www.parasoft.com

Poet Software 15
800-950-8845
www.poet.com

Pointbase 83
877-238-8798
www.pointbase.com

SilverStream 12
www.silverstream.com/extend

Sims Computing 43
406-656-7398
www.simscomputing.com

Sitraka Software 1
888-361-3264
www.sitraka.com/serverchart1/jp

Sitraka Software C3
888-361-3264
www.sitraka.com/jclass5/jp

Sitraka Software C4
888-361-3264
www.sitraka.com/jprobe3/jp

Sonic Software 48, 49
866-GET-SONIC
www.sonicsoftware.com

Sun Microsystems 6
www.sun.com/forte/ffj

Sun/Rational 57
www.rational.com/offer/sun

Togethersoft Corporation C2
919-833-5550
www.togethersoft.com

Web Builder Conference 87
800-848-5523
conferences.devx.com/webbuilder

WebGain 63
877-WEBGAIN
www.webgain.com

XML Magazine Subscriptions 76
800-848-5523
www.xmlmag.com

XML Tech Summit 64
800-848-5523
www.xmlmag.com/techsummit

Product Information

- To get immediate details on products advertised in this issue, you can simply log onto our online services at http://infoasap.devx.com.
- This provides a list of direct links to this month's advertiser's Web sites.

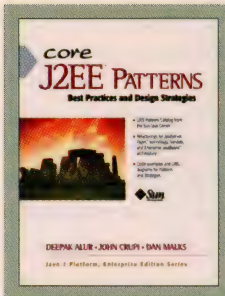
Advertising Sales Contacts

- Shana Deane, Advertising Director 413.665.6910, sdeane@fawcette.com
- Brian Levine, Eastern Regional Sales Manager 201.656.2688, blevine@fawcette.com
- Janice Mahler, Northwestern Regional Sales Manager 949.387.6684, jmahler@fawcette.com
- Chris Kourtoglou, Southern Regional Sales Manager 650.833.7155, ckourtoglou@fawcette.com

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions.
Sales Department: 209 Hamilton Avenue • Palo Alto, CA, USA 94301-2500 • 650-833-7100

Core J2EE Patterns

by Deepak Alur, John Crupi, and Dan Malks



AUTHORS: Deepak Alur, John Crupi, Dan Malks
PUBLISHER: Prentice-Hall PTR/ Sun Microsystems Press
ISBN: 0-13-066586-X
PAGES: 450
PRICE: $44.99

If you are now developing or plan to develop for the J2EE platform, buy this book! It won't teach you how to program in Java or even what JavaServer Pages (JSP), servlets, and EJB are all about, but it will make you a better J2EE developer. *Core J2EE Patterns* is a book of collected wisdom from the Sun Java Center, a group that spends their time implementing customer solutions in the real world.

The principles outlined in "Core J2EE Patterns" are based on a five-tiered model, which includes client, presentation, business, integration, and resource layers. The client may be an application or applet, or an HTML, WAP, or other browser. The presentation layer is typically implemented in JSP, servlets, or tag libraries. The business layer is typically an EJB layer. The integration layer typically handles JDBC, JMS, or other connectivity to external systems. The resource layer includes databases or any connected external system. This book focuses on the middle three layers—the presentation, business, and integration tiers.

Core J2EE Patterns is divided into three parts. Part I provides a quick introduction to the book's content and a high-level look at the J2EE platform, patterns, and best practices that the book intends to outline. After two quick chapters, you're ready to look at practical solutions. Part II centers on high-level advice about best and bad practices, along with design considerations you need to take into account. Three chapters cover the presentation and business tiers as well as refactoring advice. I found the material in these pages invaluable, putting perspective on architectural design choices I'd struggled with myself.

The bulk of the book presents J2EE patterns in Part III. Following a traditional patterns presentation format, each pattern covers context, problem, forces, solution, consequence, and sample code, along with a list of related patterns. The sample code is non-trivial and is a huge part of the book's value. The book concludes with a chapter that explores a simplified use case for a project management solution.

This book is nothing if not enlightening. Most programmers, even those with considerable experience in J2EE, have to consider a number of design choices and learn from experience. Experience, including that gained from others, can make the difference between a clean, easy to maintain system and one that quickly turns into a nightmare project. J2EE is fairly recent technology and there's nothing like useful guidelines for best practices to keep your projects on the right track. This book is an exceptional investment for any and all J2EE developers.

—Claude Duguay

Java Collections

by John Zukowski

AUTHOR: John Zukowski
PUBLISHER: Apress
ISBN: 1-893115-92-5
PAGES: 432
PRICE: $49.95

It still surprises me when I answer questions about data structures or algorithms that most Java programmers know little more than the bare minimum about the Java Collections API. A huge number of applications could be simplified or considerably enhanced, especially in terms of good performance, with a few minor changes to their choice of data structures. *Java Collections* by John Zukowski can give you the deep knowledge you need to make those more effective choices, delivering better applications to your end users.

Be aware that there is another book with the same title on the market, from Wiley. The other book takes a more academic view, covering algorithm mechanics in more detail and using the Collections API as a framework for topical discussion. It's less about the Collections API than about implementing good data structures. John Zukowski's book is targeted at programmers working with the Collections API, putting more focus on practical use and application of the APIs rather than how they could be implemented. Which book you need depends on your own objectives and interests. Of the two, I found this book much easier to read and more directly applicable to making practical implementation choices.

This book is divided into three parts. The first part looks at historical Collections, including arrays and the Vector, Stack, Enumeration, Dictionary, Hashtable, Properties, and BitSet classes, which have all been around since the earliest Java releases. Part II covers the rest of the Collections framework, including the various List, Set, and Map implementations, as well as sorting and supported array algorithms. The last few chapters in this section cover custom implementations and compatibility issues between various Java versions.

Part III looks at alternate collection libraries, including JGL, Colt, and Doug Lea's threading library. The coverage is not overly extensive in this part of the book, but it's enough to give you perspective on available choices if your needs are specific to a given area. The book ends with three appendices, a quick reference and pointers to additional information, and a meaningful discussion of generic types to be added into the Java language specification shortly.

This book covers each of the topics in a well-structured format, clearly articulating the salient points and unveiling topic matter in a logical, easy to follow order. The coverage is comprehensive and accurate—valuable for any programmer who needs to work with the Collections API. If you're serious about Java programming, get familiar with the Collections API. This book is a great vehicle for getting there.

—Claude Duguay

devX marketplace

Your #1 Source for Development Tools!

Software

Order and Download Your Favorite Tools Now!

Sitraka JProbe Profiler
Get accurate performance diagnostics, with line-by-line precision, at lightning speed.
Developer Edition
Format: Download
$799

LGB Systems XControls
Quickly develop scalable and robust database applications.
Enterprise Edition
Format: Download
$329-Save $100

nativejava.com

nativejava Jazz
Build and deploy large scale, easy to use Jazz programs with advanced functionality, and superior quality.
Format: Download
$279

ComponentOne® VSFlexGrid Pro® 7
Gives VB developers a comprehensive feature set for display and presentation of dynamic data.
Format: Download
$269-Save $30

Infragistics UltraGrid
Simplify navigation through complex data structures while providing eye-popping visual appeal.
Format: Download
$395

eHelp RoboHelp Office 9.0
Provides a powerful native environment for authoring new content—works hand-in-hand with printed documentation.
Format: Download
$899

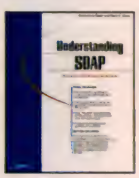
Also check online for great deals on Microsoft® products:
Microsoft® SQL Server 2000 Standard Edition
Microsoft® Access 2000
Microsoft® Outlook 2000
Microsoft® Project 2000
Microsoft

Books

Get up to speed with the best books to help you do your job better and faster.

Wireless Web Development
Understand the leading technologies in the wireless development space.
By Ray Rischpater
$27.96-Save 20%

Understanding SOAP
Create platform-independent apps using SOAP.
By Kennard Scribner and Mark C. Stiver
$31.99-Save 20%



Scripting XML and WML for Microsoft SQL Server 2000
Bring SQL Server 2000's new features into your development environment.
By Tobias Martinsson
$35.99-Save 20%

Professional BizTalk
Use BizTalk Server to integrate applications into co-operating systems.
By Stephan Mohr and Scott Woodgate
$39.99-Save 20%

Check out our complete book inventory covering:

.NET	Databases	SQL Server
Active Server Pages	Dynamic HTML	Visual Basic
ActiveX	Enterprise Development	Visual InterDev
ADO	Exchange and Outlook	Web Development
C++	Java	Windows 2000
Certification	Linux	XML
COM/DCOM	Security	

To Order:

Go online at http://marketplace.devx.com or, e-mail us at marketplace@devx.com

Please call us directly for personalized service on higher volume pricing and purchases, or for additional information.

1-888-963-3389
(outside the US call 1-650-566-2000)
8:30am - 5:30pm M-F, Pacific Time

Order Online and Save!

http://marketplace.devx.com

Or call 888.963.3389 or 650.566.2000 (outside the USA)

DevX.com MarketPlace prices are subject to change without notice. The Development Exchange, DevX, and DevX.com are trademarks of DevX.com, Inc. All products listed are registered marks or trademarks of their respective owners.

Community Spirit

by Sean **GALLAGHER**

At JavaOne, *Java Pro* held an awards ceremony to present the winners of our Readers' Choice and Java Community Awards. Last month, we gave the Readers' Choice winners the cover; this month, the Community winners get the back page.

Somehow, I think we've got this backward. But it serves as something of a metaphor for the Java community—while the products and companies get all the hype, the real work of advancing Java as a technology and as a community is done behind the scenes by developers crunching code and coming together to make things work.

> The real work of advancing Java as a technology and as a community is done behind the scenes by developers crunching code

We gave out five community awards this year. The most important of these, at least to me, was the award for Outstanding Achievement by an End-User Organization Java Developer, because I felt that we don't recognize the work done by our readers nearly enough. That award went to Yuxiang Xu of Campbell Ewald for his work as architect of the JavaServer Pages-driven Navy.com site. Naturally, he couldn't be there to accept, because he was working.

The other winners are also worthy of praise, because they have each contributed significantly to the growth of the technology we write about in this magazine every month. They are:

- Outstanding Individual Technical Contribution to the Java Community—Jaroslav Tulach, chief architect of Sun's NetBeans, for his work on the open source NetBeans Filesystems Library
- Outstanding Group Technical Contribution to the Java Community—jCorporate Open Source Team, for their work on the Expresso framework
- Outstanding Contribution to the Growth of the Java Community (Group/Organization)—iSavvix Java Community Team, for their DevSpace collaborative development site
- Outstanding Contribution to the Growth of the Java Community (Individual)—Nat Wyatt, of Informix Software, for his contributions through the development of the Cloudscape Java object-relational database.

It's no accident that three of these four awards went to what amount to open source projects, which are making considerable contributions to the advancement of software development—and to the Java platform.

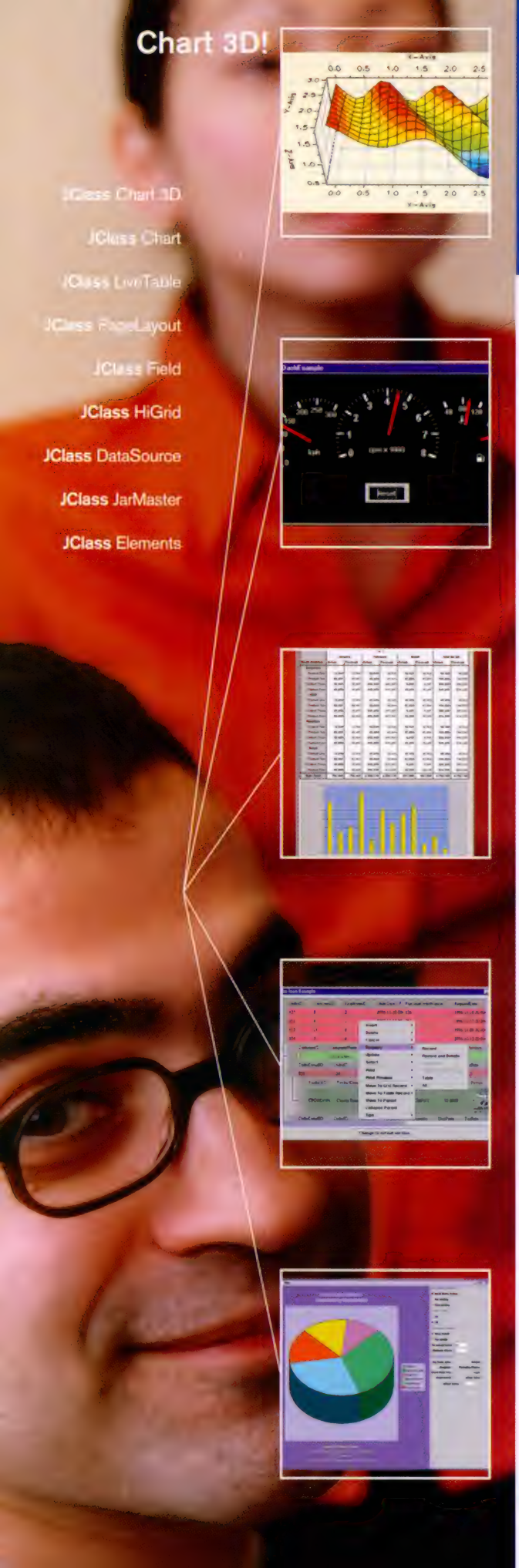
Then there's the award to Nat Wyatt. Cloudscape's fate is far from certain as I write this; there's no clear picture of what IBM intends to do with the technology, though what I've heard at this point doesn't sound promising. Considering how valuable Cloudscape has been to the Java community (it's part of the reference implementation of J2EE, for example), IBM would be foolish to squander the technology. But it's unclear from a product line standpoint how Cloudscape would integrate with, say, the DB2 family. Considering IBM's recent history of donating technologies to the Apache Software Foundation, maybe they should at least consider putting Cloudscape's code base into code escrow and donating it to the community if they fail to find a home for it at IBM or spin it off.

The Cloudscape story is an example of the challenge the Java community will face as Java becomes an increasingly mature technology. The enterprise Java market is already starting to show signs of consolidation as application server vendors are acquired, or change their focus to tools, or become increasingly marginalized. The risk is that the good technologies built by small companies will wither as they are added to the stables of the larger players and neglected.

Gartner analyst Mark Driver said at a breakfast at JavaOne, "The Java Revolution is over." He said that was a good thing, that Java had toppled the establishment and now was part of the new status quo.

That's great for vendors, but not for independent developers—status quo too often means stagnation, and the Java platform still has plenty of room for growth. The community—and you as part of it—will be responsible for ensuring that Java is not left to sit on its laurels. JP

Sean Gallagher, Editorial Director
sgallagher@fawcette.com



New in JClass™ 5.0

JClass Chart 3D™

Stunning, interactive 3D data visualization.

JClass™ 5.0

The world's leading Java components.

Develop rich and flexible client-side applications with JClass, the world's leading suite of Java components. Proven in enterprise environments around the globe, JClass Enterprise Suite includes all the functionality you need to build powerful and great-looking interfaces: charts, grids, pop-up calendars, masked edit fields, GUI extensions, JAR optimization, printing and reporting, data connectivity and more. Add to that unrivaled JDK, platform and IDE support, and you're looking at one exciting solution.

sitraka software

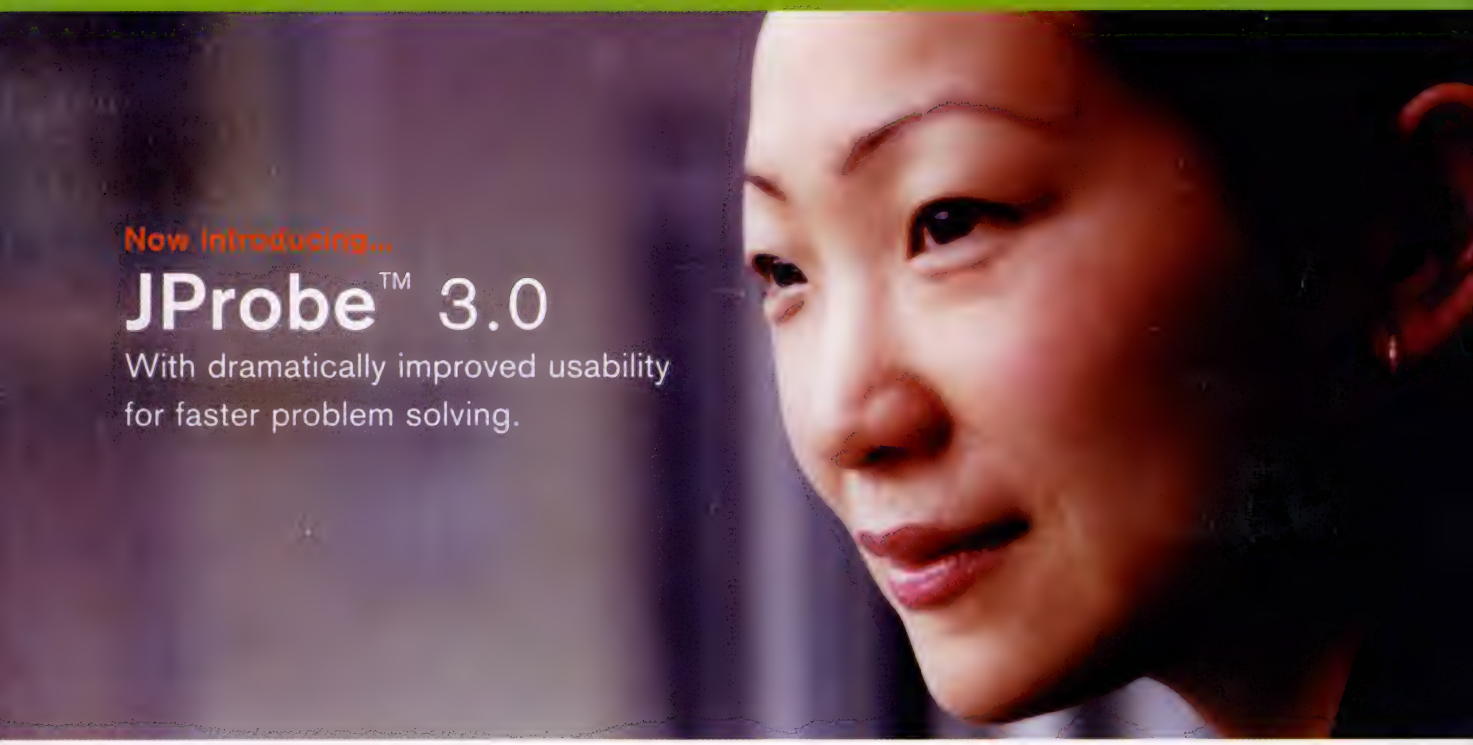
the Java™ advantage

Download a free eval now:

www.sitraka.com/jclass5/jp

1 888-361-3264 | Europe +31(0)20 510 67 00

We tune our J2EE applications with **JProbe**™. Because **performance is mission critical.**



Now introducing...

JProbe™ 3.0

With dramatically improved usability for faster problem solving.

JProbe ServerSide Suite
The ultimate solution for building performance and reliability into your server-side applications.

JProbe Profiler (with integrated Memory Debugger)
Pinpoint and eliminate performance bottlenecks and inefficient memory usage.

JProbe Threadalyzer
Track down threading hazards including deadlocks, stalls and race conditions.

JProbe Coverage
Don't let untested code put your J2EE apps at risk. Find and quantify untested code to ensure thorough and accurate test runs.

Speed. Reliability. Rock-solid code.

J2EE applications need it all. That's where JProbe comes in. JProbe integrates with all major application servers for fast, accurate analysis of your servlet, JSP and EJB code.

JProbe supports Windows, Solaris, HP UX11i and Linux.
On-site training is also available.

Visit our J2EE integration portal and download a free eval now:

www.sitraka.com/jprobe3/jp

1 888-361-3264 | Europe +31(0)20 510 67 00

sitraka
software
the Java™ advantage

© Copyright 2001 by Sitraka Inc. All rights reserved. Sitraka, the Sitraka logo and JProbe are trademarks of Sitraka Inc. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in US and other countries. All other products are trademarks or registered trademarks of their respective companies.

Bill Gates

From: Bill Gates (Microsoft)
Sent: Thursday, June 14, 2001, 11:49 AM
To: Developer & IT Professionals
Subject: Microsoft .NET Today

What will the next generation of the Internet look like? Many of us envision an online world where constellations of PCs, servers, smart devices and Internet-based services can collaborate seamlessly. Businesses will be able to share data, integrate their processes, and join forces to offer customized, comprehensive solutions to their customers. And the information you or your business need will be available wherever you are—whatever computing device, platform or application you are using.

That vision has yet to be achieved. In many respects, today's Internet still mirrors the old mainframe world. It's a server-centric computing model, with the browser playing the role of dumb terminal. Much of the information your business needs is locked up in centralized databases, served up a page at a time to individual users. Worse, Web pages are simply a "picture" of the data, not the data itself, forcing many developers back to "screen scraping" to acquire information. And integrating that underlying data with your business's existing systems—never mind those of your partners—is a costly and frustrating challenge.

Compounding this frustration is the fact that today's standalone applications and Web sites create islands of functionality and data. You have to navigate manually between Web sites, devices and applications, logging in each time and rarely being able to carry data with you. You have to keep constant track of which particular application or device or Web site gives you which level of access to which particular data. Tasks that ought to be simple—such as arranging a meeting with colleagues from partner companies and automatically updating every attendee's calendar—are a nightmare. Productivity is one of the main casualties.

Solving such problems is the key challenge for the next generation of the Internet. At the heart of the solution is eXtensible Markup Language, or XML. An open industry standard managed by the World Wide Web Consortium, XML enables developers to describe data being exchanged between PCs, smart devices, applications and Web sites. Because XML separates the underlying data from how that data is displayed, the data itself is "unlocked" so that it can easily be organized, programmed, edited and exchanged between any Web sites, applications and devices. XML is a lingua franca for the Internet age. Just as the Web revolutionized how users talk to applications, XML transforms how applications talk to each other.

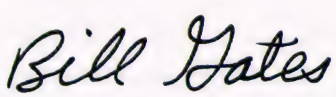
As developers become more familiar with XML, they are moving beyond simply using it for data. With the help of XML-based technologies such as SOAP (which enables applications to interoperate via standard Internet protocols) and UDDI (which gives businesses a standard way to describe their services and connect automatically), they are creating a new type of software that uses XML to provide Web-based services. These XML Web services are programmable and reusable, much like component software, except that they are accessible anywhere via the Internet. Programs using this model will run across multiple Web sites, drawing on information and services from each of them, and combining and delivering them in customized form to any device.

How will businesses and their customers benefit from this? Because XML Web services break down the distinctions between the Internet, standalone applications and computing devices of every kind, they enable businesses to collaborate to offer an unprecedented range of integrated and customized solutions—solutions that enable their customers to act on information any time, any place and on any device.

The power of the XML Web services model is amazing. A company offering an online electronic-payment service can expose its service to partners, so that they can deliver it as part of their own offering—regardless of what platform they are using. An airline can link its online reservation system to that of a car-rental partner, so travelers can book a car at the same time they book a flight. An online auction company can notify bidders when they are outbid or have won an auction, or could partner with other firms to offer alternative shipping, fulfillment or payment options. XML Web services help your business break free of its boundaries.

With XML Web services gaining momentum among developers as the next generation of Internet-based computing, it's time to deliver a platform that makes it simpler to build these solutions and provides a reliable framework for integration and interoperability. Such a platform must be based on open standards, so it can work across all programming languages, operating systems and applications. And it must combine the power of PCs and smart devices with the richness of the Internet.

Microsoft's platform for building, deploying, operating and integrating XML Web services is .NET. In the next few pages we'll describe .NET, explain its many benefits, and set out a roadmap for transforming your business to take advantage of everything it has to offer.



Chairman and Chief Software Architect

As a result of the changes in how businesses and consumers use the Web, the industry is converging on a new computing model that enables a standard way of building applications and processes to connect and exchange information over the Web. This new Internet-based integration methodology, called "XML Web services," enables applications, machines, and business processes to work together in a way never previously possible. The widespread support around XML within the developer community assures that businesses will thrive and cooperate in the accelerated, Internet-based economy with this XML Web services model.

The first principle of XML Web services is that systems connect through the Internet—a safe assumption given the high availability and low-cost connectivity provided by the Internet. Second, there needs to be a simple way to "find" services on the Internet with which businesses can work. UDDI (Universal Description, Discovery, and Integration — www.uddi.org) is a broad industry effort, involving companies such as Microsoft, Ariba, COMPAQ, Dell, HP, IBM, SAP, and many others, which provides a way to locate and understand services provided by other companies. It's a kind of "yellow pages" on the Internet for the industry.

Third, a common language is needed to ensure that information is shared with others. XML is the ingredient that makes this possible. XML provides a common data format so that you can continue to work with data and information in a way that doesn't require business partners or customers to use a particular programming language, application, or operating system to interact with your systems. The final principle is that there must be a way to actually conduct business—for example, to call the service, book the appointment, order the part, or deliver the information—through a common protocol. This protocol is called SOAP (Simple Object Access Protocol). It enables systems to talk to one another and make requests. SOAP is the new model for Internet native integration.

These four principles enable you to connect, find, transform, and transact across systems, applications, and processes to deliver XML Web services. XML Web services are flexible technologies that bind disparate systems across different languages, unifying personal computing, enterprise computing, and the Web. As long as the fundamental communication occurs via XML Web services, each system can be independent from the others, with each "service" running on entirely different systems, even in different parts of the world.

The benefits of XML Web services are far-ranging and will be instrumental in propelling explosive business growth over the next few years. Businesses will be able to:

Easily integrate with other businesses using XML Web services. Your software will easily integrate with other pieces of software—from the desktop to the mainframe—both within your enterprise and at external sites. These integration capabilities enable you to forge closer ties with business partners and pursue best-of-breed integration of business processes.

Develop applications faster. As the pool of XML Web services grows, developers will be delivering more and more software and services, including legacy applications, that operate within the XML Web services programming model. If there is existing code that can help you solve your problem, you can find it and integrate with it through XML Web services, instead of reinventing it.

Easily implement personalization. Integration with external data sources is part of the XML Web services programming model. This makes it possible for you to request information and transform data by whatever means necessary to deliver individualized software and services. Your maintenance burden is reduced because you no longer have to collect and maintain this information on an application-by-application basis.

A New User Experience. Employing XML Web services provides benefits for both consumers and businesses.

Consumers will enjoy unparalleled ease of use when they use applications built with XML Web services. Because XML Web services link applications, services, and devices together as connected solutions, software arrives as part of an integrated experience that offers simplicity in computing. XML Web services give users the ability to act on information any time, any place, from any smart device.

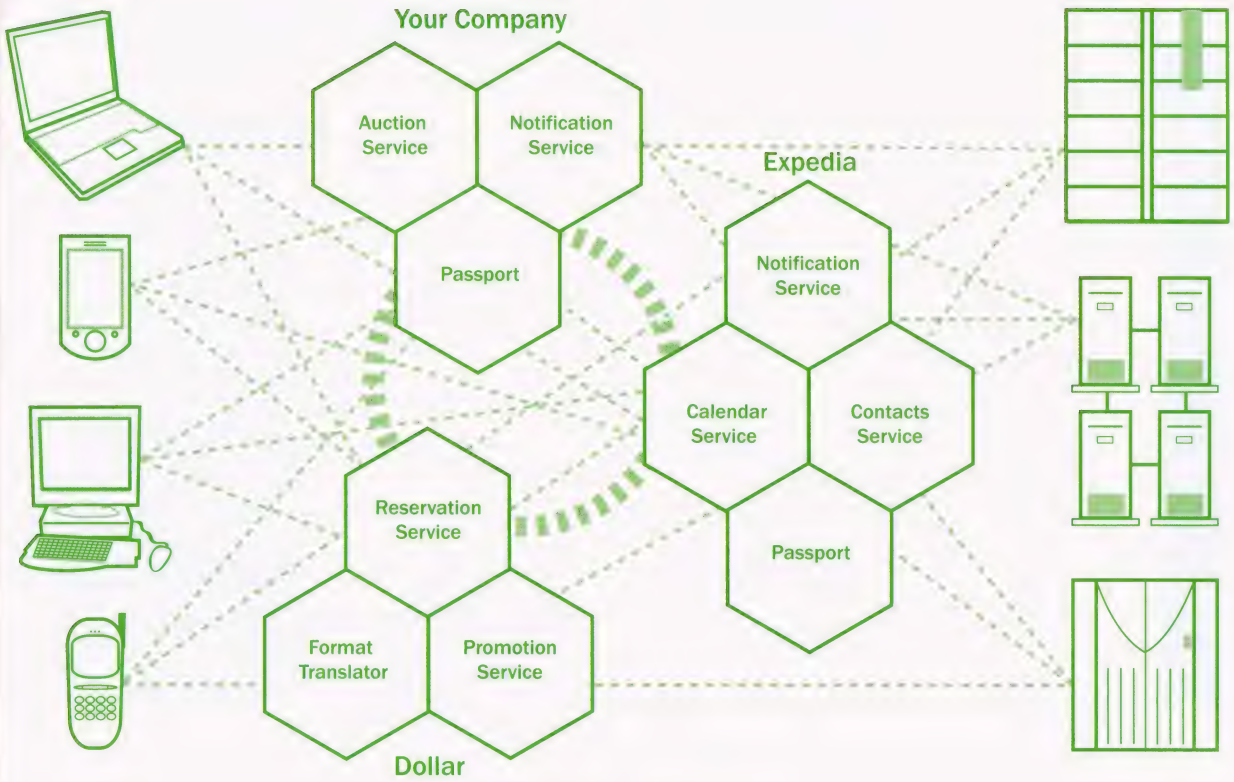
For businesses, implementing software projects using XML Web services solves many of today's supply-chain and demand-chain integration challenges. With its industry-standard communication interfaces, an XML Web service is a simple, reliable way to blend existing systems with new applications and services.

Microsoft .NET—
An XML Web Services Platform

With the momentum behind XML Web services growing among developers, Microsoft is building a platform to help them write, deploy, and manage these services. Microsoft® .NET is that platform. It contains both products and

A Revolution Is Upon Us. Revolutions are a way of life in the computer industry. Only 20 years ago, the world was still in the mainframe era. Few people had access to or used computers, and when they did, it was only through the nearest IT department. The PC, the graphical user interface, and the introduction of the Internet changed all that. They democratized computing for hundreds of millions of people and transformed the computer into a mass-market product.

Since then, standards such as HTML and HTTP have exponentially increased people's use of the Internet. This base protocol for viewing content on the Web (and the associated software for "browsing" this content) grew Web usage to what it is today—a key activity in the daily lives of business employees and consumers.



XML Web services can be leveraged in endless combinations to create User Experiences.

services developers need to build and run XML Web services, and enables some of the User Experiences that demonstrate the integration of XML Web services with the universe of computers, smart devices, and Web-based services used by consumers.

Microsoft .NET is optimized for XML and XML Web services, but it is by no means the only choice of development platform. Because XML is an open standard, created and managed by the World Wide Web Consortium with the input of many computer-industry companies, the products and services within Microsoft .NET will interoperate with a broad set of XML- and XML Web services–enabled infrastructure and developer tools from other vendors. With over 5 million developers skilled on Microsoft technology, the development community is poised to begin mass delivery of XML Web services.

DOLLAR MARKET FACT

In two weeks, programmers built, tested, and deployed a Microsoft solution that translates reservation requests and data between Dollar® Rent-A-Car's VMS-based reservation systems and an airline partner's UNIX servers. Because XML Web services powered the solution, Dollar can reuse that same integration model to link with any number of partners.

MICROSOFT .NET PRODUCTS AND SERVICES

Developer tools and technologies. A productive set of tools is critical to developer success on a new platform like .NET. Visual Studio.NET and the Microsoft .NET Framework supply a complete solution for building, deploying, and running XML Web services. They help you maximize the performance, reliability, and security of your XML Web services.

Visual Studio.NET is the next generation of Microsoft's popular multi-language development tool, built especially for .NET. Visual Studio.NET helps developers quickly build XML Web services and applications that scale easily, using the language of their choice. Visual Studio.NET advances the following high-productivity programming languages: Visual Basic®, which includes new object-oriented programming features; Visual C++®, which advances Windows®-based development and enables you to build .NET applications; and C#, which brings RAD to the C and C++ developer. In addition to these languages provided by Microsoft, there will be over 20 languages provided by partners, including Perl,

Python, Cobol, and Eiffel, from which developers can choose.

The .NET Framework is a high-productivity, standards-based, multi-language application execution environment that handles essential plumbing chores and eases deployment. It provides an application execution environment that manages memory, addresses versioning issues, and improves the reliability, scalability, and security of your application. The .NET Framework consists of several parts, including the Common Language Runtime, a rich set of class libraries for building XML Web services, and ASP.NET, the next generation of Active Server Pages.

Server infrastructure. XML Web services should be built on a next-generation infrastructure that offers developers the benefits of modular architecture, economical and linear scaling, security, reliability, manageability, and high availability. The .NET Enterprise Servers and the Windows 2000 Server family make up the Microsoft .NET server infrastructure for deploying, managing, and orchestrating XML Web services. Designed with mission-critical performance in mind, they provide enterprises with the agility they need to integrate their systems, applications, and partners through XML Web services, and the flexibility to adapt to changing business requirements.

The Windows 2000 Server family is Microsoft's secure, scalable foundation for running the .NET Enterprise Servers and the next generation of business applications.

The .NET Enterprise Servers are:

- **Application Center 2000** to deploy and manage highly available and scalable Web applications;
- **BizTalk™ Server 2000** to build XML-based business processes across applications and organizations;
- **Commerce Server 2000** for quickly building scalable e-commerce solutions;
- **Content Management Server 2001** to manage content for dynamic e-business Web sites;
- **Exchange 2000 Server** to enable messaging and collaboration, anytime, anywhere;
- **Host Integration Server 2000** for bridging data and applications on legacy systems;
- **Internet Security and Acceleration Server 2000** for secure, fast Internet connectivity;
- **Mobile Information 2001 Server** to enable application support by mobile devices like cell phones;
- **SharePoint™ Portal Server 2001** to find, share, and publish business information; and
- **SQL Server™ 2000** to store, retrieve, and analyze structured XML data.

Services. An XML Web services model offers developers an opportunity to achieve economies of scale by utilizing a set of core XML Web services that releases developers from the burden of building everything themselves. By integrating with these core services to perform routine tasks, developers can concentrate on building high-value, business-critical XML Web services. Microsoft Passport is the first such service, providing authentication services for over 160 million accounts. Microsoft and many other companies plan to develop more core services for users and enterprises.

EXPEDIA MARKET FACT

XML Web services will help Expedia.com® transform itineraries into communication centers—allowing travelers to pick distinct notification settings for different members of their integrated contact list. By providing customers with relevant and timely information at their convenience, Expedia will continue to attract new customers and further strengthen the loyalty of their existing customer base.

Clients. Microsoft will support XML Web services across the full range of smart clients so that your customers can access your XML Web services regardless of location or type of device. Some of the clients Microsoft offers include Windows CE, Windows Embedded, Windows 2000, and the upcoming Windows XP. These clients will power PCs, laptops, workstations, smart phones, handheld computers, Tablet PCs, Xbox™ game consoles, and many other smart devices.

User Experiences. A good User Experience pulls together all of the XML Web services and client software a particular user needs, and presents everything to the user in an integrated way that makes sense—even if it calls on XML Web services run by other companies on other machines. Microsoft will deliver User Experiences for knowledge workers, consumers, enterprises, small businesses, and developers. Some of the products that Microsoft is transitioning into User Experiences are Microsoft Office, MSN®, bCentral™, and Visual Studio.NET.

YOUR PATH TO XML WEB SERVICES

Microsoft is delivering the product line to put your business on the path to XML Web services today:

- **Windows 2000 Server**—the reliable infrastructure for the linear, cost-effective scaling required by the XML Web services world.
- **The .NET Enterprise Servers**—the foundation for running, managing, and securing XML Web services.
- **XML Web services**—Passport.
- **XML Web service–capable clients**—Windows 2000, Windows Me, Windows CE, Office XP, and soon Windows XP.

XML-Enabled Products & Services		PRESENT		FUTURE
INFRASTRUCTURE	Client Operating Systems	Windows 2000 Professional Windows CE	Windows Me	Windows XP Professional Windows XP Home Edition Windows XP Embedded Windows CE "Talisker"
PCs AND SMART DEVICES	Smart Devices	Pocket PC	Mobile Explorer™	Xbox Tablet PC Smart phone code-named "Stinger" UltimateTV®
PCs AND SMART DEVICES	User Experiences	MSN Explorer Office XP	Visio® 2002	Next version of Office Next version of Microsoft Project Next version of Visio
WEB SERVICES	Building Block Services	Passport		"HailStorm" Services
WEB SERVICES	Developer Tools	Visual Studio 6.0 SOAP Toolkit 2.0	Visual Studio.NET Beta 2 .NET Framework Beta 2	Visual Studio.NET .NET Framework .NET Compact Framework
INFRASTRUCTURE	Servers	Windows 2000 Server Windows 2000 Advanced Server Windows 2000 Datacenter Server SQL Server 2000 Exchange 2000 Server Commerce Server 2000 Host Integration Server 2000	ISA Server 2000 BizTalk Server 2000 Application Center 2000 SharePoint Portal Server 2001 Mobile Information 2001 Server Content Management Server 2001	Next version of Windows code-named "Whistler" Server family Next version of SQL Server code-named "Yukon"

- **Developer tools and technologies**—the latest version of the Microsoft SOAP Toolkit for XML Web services and betas of Visual Studio.NET.

Leading-edge IT organizations, like **Dollar Rent-A-Car**, **Expedia.com**, and others, use these products today to build high-value XML Web services that solve today's integration and interoperability problems. For example, when CheckSpace offered its electronic payment processing capabilities through an XML Web service, it expanded its market reach to new small-business customers, resellers, and online marketplaces that used a variety of computer systems and accounting packages. Those new customers now enjoy streamlined accounting processes and more efficient cash-flow management.

FIVE EASY STEPS

Now is the time to prepare your company to take advantage of XML Web services. Five easy steps lead to the future:

1. Educate yourself and your IT department about XML Web services and Microsoft .NET;
2. Investigate Windows 2000, Office XP, and the Microsoft .NET Enterprise Servers as a way to upgrade the XML capabilities of your systems and infrastructure;
3. Download and evaluate betas of Microsoft's developer tools;
4. Create pilot projects that test XML Web services; and
5. Insist that your vendors have a roadmap for making their applications accessible as XML Web services.

A subscription to **MSDN® Universal** delivers the developer tools and servers you need to learn about .NET. Microsoft also offers .NET Readiness Training, which teaches the technical foundation for creating XML Web services and instructs you on using tools like C# and Visual Studio.NET.

Get started and learn more. For more information about XML Web services and Microsoft .NET, visit **microsoft.com/net**

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 2001 Microsoft Corporation. All rights reserved. Microsoft, bCentral, BizTalk, Mobile Explorer, MSDN, MSN, SharePoint, UltimateTV, Visio 2002, Visual Basic, Visual C++, Visual Studio, Windows, and Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

One Year ONLY $19.95!

74% off the cover price of JAVAPro

Name:

Company Name:

Address:

City State: Zip:

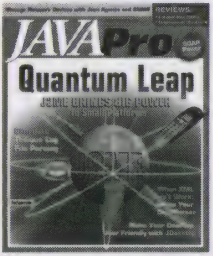
Country:

E-mail (required):

One year of *Java Pro* is 12 issues plus one special issue. Savings based on the annual cover price of $77.35. Please allow four-to-six weeks for delivery of the first issue. International subscriptions must be prepaid in U.S. funds plus postage: Canada and Mexico add $18 per year for surface delivery; all other countries add $44 per year for airmail delivery. New orders only.

☐ YES!

Start my subscription to *Java Pro* and bill me for just $19.95



JAVAPro

49JF

Subscribe Online: http://www.javapro.com

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVAPro™

PO Box 59277
Boulder, CO 80322-9277

☐ YES, please call me, I need a total business integration solution that fits my company's unique e-business requirements.

Please send me:

☐ A FREE copy of IONA's Whitepaper:
"The Path to Portals" which discusses the merits and benefits of enterprise portals and explains how IONA is positioned to help me maximize my IT investment.

☐ Information on other e-business integration products from IONA.

Please fill in the following information:

Name:_____

Company Name:_____

Address:_____

City, State, Zip:_____

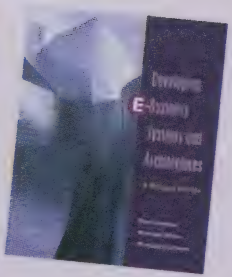
Phone:_____

FAX:_____

E-mail:_____

(required to e-mail whitepaper.)

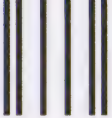
AD0401EBSBRC



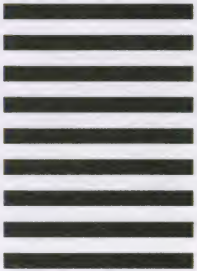
Introducing your FREE copy of "Developing E-Business Systems and Architectures – A Manager's Guide", *written by Guttman, Harmon and Rosen.*

For your FREE COPY, return this business reply card, reply online at www.iona.com/psarchitecture or call us at 1-800-672-4948.

IONA



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO.11444 WALTHAM, MA

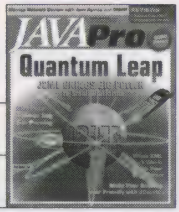
POSTAGE WILL BE PAID BY ADDRESSEE

IONA TECHNOLOGIES INC
200 WEST ST
WALTHAM MA 02451-9901

One Year ONLY $19.95!

JAVA Pro

74% off one year of Java Pro!



Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

YES!

Start my subscription to *Java Pro* and bill me for just $19.95

One year of *Java Pro* is 12 issues plus one special issue. Savings based on the annual cover price of $77.35. Please allow four-to-six weeks for delivery of the first issue. International subscriptions must be prepaid in U.S. funds plus postage: Canada and Mexico add $18 per year for surface delivery; all other countries add $44 per year for airmail delivery. New orders only.

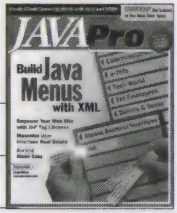
49JD

Subscribe Online: http://www.javapro.com

One Year ONLY $19.95!

JAVA Pro

74% off one year of Java Pro!



Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

YES!

Start my subscription to *Java Pro* and bill me for just $19.95

One year of *Java Pro* is 12 issues plus one special issue. Savings based on the annual cover price of $77.35. Please allow four-to-six weeks for delivery of the first issue. International subscriptions must be prepaid in U.S. funds plus postage: Canada and Mexico add $18 per year for surface delivery; all other countries add $44 per year for airmail delivery. New orders only.

49JD

Subscribe Online: http://www.javapro.com

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVAPro™

PO Box 59277
Boulder, CO 80322-9277

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

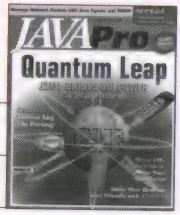
POSTAGE WILL BE PAID BY ADDRESSEE

JAVAPro™

PO Box 59277
Boulder, CO 80322-9277

One Year ONLY $19.95! JAVAPro

74% off one year of Java Pro!



Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

YES!

Start my subscription to *Java Pro* and bill me for just $19.95

One year of *Java Pro* is 12 issues plus one special issue. Savings based on the annual cover price of $77.35. Please allow four-to-six weeks for delivery of the first issue. International subscriptions must be prepaid in U.S. funds plus postage: Canada and Mexico add $18 per year for surface delivery; all other countries add $44 per year for airmail delivery. New orders only.

49JN

Subscribe Online: http://www.javapro.com

Do you need a cutting edge tool to enhance your web site?

Subscribe to WEBBuilder CD

One Year ONLY $29.95!

Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

YES!

Send me one year (4 issues) of *Web Builder CD* and bill me for just $29.95

*One year 4, printed quarterly. Savings based on the annual cover price of $51.80. Subscription begins upon receipt of payment. Allow four-to-six weeks for delivery of the first issue. International subscriptions must be payable in U.S. dollars plus postage: Canada and Mexico add $8 per year for surface delivery; all other countries add $15 per year for airmail delivery. Canadian GST included. New Orders Only.

49JN

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVAPro

PO Box 59277
Boulder, CO 80322-9277

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

WEBBuilder

WBCD
PO Box 15025
North Hollywood, CA 91615-9953

FREE SUBSCRIPTION TO XML MAGAZINE!

BUILDING THE NEXT GENERATION INTERNET

❑ **YES!** I would like to receive *XML Magazine* free of charge!

❑ **No** I would not like to receive *XML Magazine* free of charge!

Signature *(required)*

Date *(required)*

Name

Title

Company

Address

City State Zip Country

Phone Fax

E-mail Address *(required)*

❑ Please do not use my email to contact me about my subscription status and related Fawcette products 49JN

❑ Please do not send me third party email messages about products and services from *XML Magazine* partners

Subscriptions are free to qualified IS Professionals in the U.S. Paid subscriptions are available for non-U.S. readers. International subscriptions are to be prepaid in U.S. funds; Canada/Mexico $64.00, all other $84.00.

For faster service renew online at www.xmlmag.com/free

or fax to: 925/372-8582

Please answer all questions, sign and date. Incomplete forms cannot be processed.

1. PRIMARY BUSINESS ACTIVITY AT YOUR LOCATION (SELECT ONLY ONE):

101 ❑ Finance, banking, accounting, insurance
102 ❑ Manufacturing other than computer hardware and software and software
103 ❑ Manufacturing computers and computer software
104 ❑ Non-computer retailer, distributor, wholesaler
105 ❑ Applications software development
106 ❑ Aerospace
107 ❑ Government, military
108 ❑ Agriculture, mining, construction, oil & gas
109 ❑ Transportation, utilities
110 ❑ Media, marketing, advertising
111 ❑ Real estate
112 ❑ Research and development
113 ❑ Engineering
114 ❑ VAR/Systems Integrators
115 ❑ Communications services
116 ❑ Computer-related retailer, distributor, wholesaler
117 ❑ Systems software development
118 ❑ Education
119 ❑ Health, medical, legal
120 ❑ Consulting (please specify)

199 ❑ Other (please specify)

2. WHAT IS YOUR CURRENT JOB TITLE? (SELECT ONLY ONE FROM THE FOLLOWING SIX (6) CATEGORIES):

Application development manager/ architect:

201 ❑ Applications development manager
202 ❑ Software development director
203 ❑ Software architect
204 ❑ Software project leader
205 ❑ Project manager
206 ❑ Advanced technology evaluator
207 ❑ Software team leader
208 ❑ Software engineering supervisor
209 ❑ Systems analyst/designer
210 ❑ Database designer
211 ❑ Application manager/administrator
295 ❑ Other (please specify)

IS manager:

230 ❑ CIO, Technical VP
231 ❑ IS/MIS manager
296 ❑ Other (please specify) 4/01

Over Please ☞

Application developer:

240 ❑ Applications developer/ programmer
297 ❑ Other (please specify)

Computer operations staff:

250 ❑ System Administrator
251 ❑ LAN Administrator
252 ❑ Help Desk
298 ❑ Other (please specify)

Corporate manager:

260 ❑ CEO, president, partner, chairman, owner
261 ❑ VP, controller, director of marketing/sales, general manager, division/department head

Other (categories not listed above):

299 ❑ Other (please specify)

3. HOW ARE YOU CURRENTLY USING XML?

301 ❑ Currently using
302 ❑ Planning to use XML in the next 12 months
303 ❑ Evaluating XML for applicability
304 ❑ Not planning on using XML

4. ARE YOU DEVELOPING APPLICATIONS THAT WILL INVOLVE ONE OR MORE SERVERS OR HOSTS OR DO YOU PLAN TO IN THE NEXT 12 MONTHS?

401 ❑ Yes
402 ❑ No

5. SERVER/HOST OPERATING SYSTEMS IN USE AT THIS ENTIRE LOCATION (PLEASE CHECK ALL THAT APPLY):

501 ❑ Windows NT/2000 Server
502 ❑ Unix (all flavors)
503 ❑ Netware/Intranetware
504 ❑ OS/400
505 ❑ VMS
506 ❑ MVS/VM
507 ❑ OS/2
599 ❑ Other (please specify)

6. WHAT IS THE ANNUAL DOLLAR VALUE OF COMPUTER PRODUCTS AND SERVICES FOR WHICH YOU ARE PERSONALLY INVOLVED IN THE PURCHASING DECISIONS FOR YOUR ORGANIZATION'S OR LARGEST CLIENT'S INTERNAL USE?

601 ❑ More than $10 million
602 ❑ $5 million - $9,999,999
603 ❑ $2 million - $4,999,999
604 ❑ $1 million - $1,999,999
605 ❑ $500,000 - $999,999
606 ❑ $100,000 - $499,999
607 ❑ $50,000 - $99,999
608 ❑ $10,000 - $49,999
609 ❑ Less than $10,000

7. WHAT IS THE NUMBER OF EMPLOYEES OF THE LARGEST BUSINESS ENTITY BY WHOM YOU ARE EMPLOYED?

701 ❑ Under 100
702 ❑ 100 - 999
703 ❑ 1,000 - 4,999
704 ❑ 5,000 - 9,999
705 ❑ 10,000 - 49,999
706 ❑ More than 49,999

Free subscription

FOLD AND TAPE, DO NOT STAPLE

PO Box 2507
Martinez, CA 94553-9917

XML MAGAZINE

POSTAGE WILL BE PAID BY ADDRESSEE

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

Do you need a cutting edge tool to enhance your web site?

Subscribe to WEBBuilder CD

One Year ONLY $29.95!

Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

*One year 4, printed quarterly. Savings based on the annual cover price of $51.80. Subscription begins upon receipt of payment. Allow four-to-six weeks for delivery of the first issue. International subscriptions must be payable in U.S. dollars plus postage: Canada and Mexico add $8 per year for surface delivery; all other countries add $15 per year for airmail delivery. Canadian GST included. New Orders Only.

❑ YES!

Send me one year (4 issues) of *Web Builder CD* and bill me for just $29.95

49JD

One Year ONLY $29.95!

❑ YES! **Send me one year (4 issues) of *Web Builder CD* and bill me for just $29.95**

WEBBuilder CD

Name:

Company Name:

Address:

City State: Zip:

Country:

E-mail (required):

42% OFF

*One year 4, printed quarterly. Savings based on the annual cover price of $51.80. Subscription begins upon receipt of payment. Allow four-to-six weeks for delivery of the first issue. International subscriptions must be payable in U.S. dollars plus postage: Canada and Mexico add $8 per year for surface delivery; all other countries add $15 per year for airmail delivery. Canadian GST included. New Orders Only.

49JD

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

WBCD
PO Box 15025
North Hollywood, CA 91615-9953

NO POSTAGE
NECESSARY IF
MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 PALO ALTO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

WBCD
PO Box 15025
North Hollywood, CA 91615-9953